Content Recommendation Through Linked Data

ScuDo

Scuola di Dottorato ~ Doctoral School

WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation

Doctoral Program in Computer and Control Engineering (29^{th}cycle)

Content Recommendation Through Linked Data

By

Iacopo Vagliano

******

Supervisor(s):

Prof. Maurizio Morisio

Doctoral Examination Committee:

Dr. Fabien Gandon, Referee, Inria, CNRS, I3S

Prof. Andrea G. B. Tettamanzi, Referee, Université Côte d'Azur, Inria, CNRS, I3S

Prof. Fulvio Corno, Politecnico di Torino

Prof. Paweł Czarnul, Gdansk University of Technology

Prof. Marco Torchiano, Politecnico di Torino

Politecnico di Torino

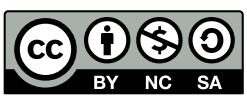
2017

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Iacopo Vagliano
2017

This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).



To my dad

"It's not information overload. It's filter failure."

Clay Shirky

Acknowledgements

With this thesis, a season of my life is reaching the end. I expected science to be about answers, and I discovered it is much more about questions. Actually, the more we learn, the more we understand how much we still need to explore; and if we exactly knew what we were doing, we would not call it research.

Firstly, I would like to thank my supervisor, Maurizio Morisio. I appreciate all the trust and freedom that I was granted in conducting my research, the valuable discussions and advices, and the chance to deal with situations which I though beyond my capabilities at the time. I am also grateful for the means provided by TIM (formerly Telecom Italia) to support my research. In particular, thanks to my second supervisor, Marco Marengo, your open mind, pragmatism, and interest in a real collaboration between industry and academia were rich stimuli for me.

I would like to express sincere gratitude to all the colleagues of the SoftEng group. First of all, thanks to Cristhian Figueroa and Oscar Rodriguez for the fruitful collaboration, which motivated me a lot in these years. I am also grateful to Marco Torchiano for his availability and for sharing his experience. I also appreciated the precious help in technical subjects and the hints to deal with the bureaucratic issues received by Luca Ardito. Indeed, I would like to mention all the mates who made the atmosphere in the Lab 1 more pleasant: Rifat Rashid, Erion Cano, Riccardo Coppola, Diego Monti, Francesco Strada, Amirhosein Toosi and Alysson Dos Santos. Of course, it was also my pleasure to collaborate (and play soccer table in the breaks) with the other present and past members of JOL MobiLab group. So, thanks to Lucia Longo, who was responsible of the TIM support in the final period of my Ph.D. program, Eleonora Gargiulo, Gianluca Cecchi, Mirko Rinaldini, Alessandro Izzo, Luisa Rocca, Enrico Catalano, and all the other interns and undergraduates who joined the group for a while in these years.

An outstanding experience during these years was the research visiting at the Gdansk University of Technology in Poland. I would like to thank all the colleagues of the Knowledge Management group, in particular, Krzysztof Goczyła, which supervised me, Wojciech Waloszek for the useful discussions, and Aleksandra Karpus, for the results reached through our joint effort. I am grateful also to Andrzej Wardziński, Aleksander Jarzębowicz and Jakub Miler for guesting me in their office.

A special thank to the "train guys": Mattia Berardo, Marco Basso, Piergianni Serra, Paolo Arnolfo, and, occasionally, Marco Fanti, who have been my mates in the everyday Cuneo-Torino travels (in the good and in the bad) for two years. We have found a nice trade-off between funny and serious chat and, for this reason, we have been hated by all those who were just wishing to sleep in the early morning. On the contrary, Marco Conoscenti suffered me for about one year as a flatmate at "Cumiana, 44". He found time for suggesting me intellectual books, having ambitious discussions about everything and nothing, and above all, offering me delicious Sicilian food.

Finally, I cannot forget my strong foundation: my family and Zosia. Thanks for your unavoidable support. Zosia, our path together started at the same time than the Ph.D. program, but in you, I discovered something much more precious than any qualification or scientific discovery.

Abstract

Nowadays, people can easily obtain a huge amount of information from the Web, but often they have no criteria to discern it. This issue is known as information overload. Recommender systems are software tools to suggest interesting items to users and can help them to deal with a vast amount of information. Linked Data is a set of best practices to publish data on the Web, and it is the basis of the Web of Data, an interconnected global dataspace.

This thesis discusses how to discover information useful for the user from the vast amount of structured data, and notably Linked Data available on the Web. The work addresses this issue by considering three research questions: how to exploit existing relationships between resources published on the Web to provide recommendations to users; how to represent the user and his context to generate better recommendations for the current situation; and how to effectively visualize the recommended resources and their relationships.

To address the first question, the thesis proposes a new algorithm based on Linked Data which exploits existing relationships between resources to recommend related resources. The algorithm was integrated into a framework to deploy and evaluate Linked Data based recommendation algorithms. In fact, a related problem is how to compare them and how to evaluate their performance when applied to a given dataset. The user evaluation showed that our algorithm improves the rate of new recommendations, while maintaining a satisfying prediction accuracy. To represent the user and their context, this thesis presents the Recommender System Context ontology, which is exploited in a new context-aware approach that can be used with existing recommendation algorithms. The evaluation showed that this method can significantly improve the prediction accuracy. As regards the problem of effectively visualizing the recommended resources and their relationships, this thesis proposes a

visualization framework for DBpedia (the Linked Data version of Wikipedia) and mobile devices, which is designed to be extended to other datasets.

In summary, this thesis shows how it is possible to exploit structured data available on the Web to recommend useful resources to users. Linked Data were successfully exploited in recommender systems. Various proposed approaches were implemented and applied to use cases of Telecom Italia.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the years, the amount of information generated on the Web has exploded. To have an idea of the enormity of this explosion, every minute over 3 million likes occur, roughly 243 thousand new photos are uploaded, and more than 3 million posts are shared on Facebook. At the same time, around 56 thousand pictures are uploaded on Instagram; about 430 thousand Tweets are published; 300 hours of video are uploaded to Youtube, and 120 new accounts are opened on LinkedIn.[1] This means that nowadays people can easily obtain an enormous amount of information from the Internet, but often they have no criteria to discern it. This issue is known as information overload.

Besides traditional search engines, researchers have developed more intelligent tools to help the user deal with an enormous amount of information, such as Recommender Systems (RS), which are software tools that suggest interesting items to the user [1]. At the same time, the Web has evolved from an information space for sharing textual documents into a medium for publishing structured data. Linked Data[2] is a set of best practices to publish and interlink data on the Web, and it is the basis of the Web of Data, an interconnected global dataspace where data providers publish their content publicly. The Web of Data consists of a huge knowledge repository containing different kind of information varying from encyclopedic and linguistic to real-time (e.g. data streams) and user generated content. Moreover, the key feature is the embedding of semantic relationship between the entities represented.

[1] Statistics published in May 2016 available at http://www.go-globe.com/blog/60-seconds/
[2] http://linkeddata.org

The idea of introducing semantics into RS is not new, and many works have been proposed before the Linked Data was conceived [2–9]. Most of these approaches exploited specific domain ontologies and taxonomies to support traditional techniques and addressed cold-start and data sparsity, two well-known problems of RS. However, they are not particularly suited to deal with datasets in the Web of Data, and new methods are required to incorporate Linked Data into RS by effectively exploiting their semantics [10]. The main reason to provide new approaches is that ontological RS relies on closed domain ontologies defined ad hoc, which often require an high maintenance effort, while the Web of Data is based on the open world assumption and data models may change rapidly since vocabularies and ontologies used in it are designed to be extended easily. Additionally, datasets in the Web of Data are published according to the Linked Data principles by using the Resource Description Framework (RDF)[3] [11] data model, which represents information in a graph form (and is shortly described in Section 2.3.2). Thus, they require specific paradigms to be integrated into RS.

Di Noia and Ostuni [10] summarized the main benefits that Linked Data can provide to RS. Firstly, it consists of an enormous amount of multi-domain and ontological information which is freely available. Secondly, it provides standard access to data. Finally, it represents semantic relationships among different entities which are already structured, interlinked and based on ontologies. Besides, Linked Data can decrease the dependency from the user since its interlinked nature enables content-based recommendations. For example, we could combine a Linked Data based approach with a classical recommendation method that exploits user ratings to mitigate the cold-start problem, which occurs when the system need to recommend items to a new user (namely, a user that have still not rated any item). Moreover, multilingual datasets as DBpedia[4] [12], the Linked Data version of Wikipedia, can enable cross-language applications as Narducci et al. [13] have shown.

This thesis discusses how to discover useful information for the user from the vast amount of structured data, and notably Linked Data available on the Web. In particular, the reference scenario, which summarizes the primary needs of Telecom Italia, is the following: a mobile user is coming back from work (for example, she is waiting for the bus or walking down the street) and wants to decide which movie to watch tonight. She may focus on this activity for a limited amount of time (soon

[3] http://www.w3.org/standards/techs/rdf
[4] http://dbpedia.org/

the bus may arrive, or she may reach the destination she is walking to) and wants to explore on the Web some information about movies she may like, rather than read details about a specific movie. In this context, this thesis poses the following research questions:

RQ1 *How can existing relationships between resources published on the Web be exploited to provide recommendations to users?*

RQ2 *How can the user and her context be represented to generate better recommendations for the current situation?*

RQ3 *How can the recommended resources and their relationships be effectively visualized?*

Regarding RQ1, Linked Data may improve RS because they represent multi-domain knowledge, provide standard access to data, and represent semantic relationships among different entities, as previously explained. This thesis proposes a new algorithm based on Linked Data which exploits existing relationships between resources to recommend related resources. It dynamically analyzes the categories they belong to and their explicit references to other resources, then combines the results. The algorithm has been integrated into a framework to deploy and evaluate Linked Data based recommendation algorithms. In fact, a related problem is how to compare Linked Data based algorithms and how to assess their performance when applied to a particular dataset in the Web of Data. The user evaluation compared the proposed algorithm with state-of-the-art algorithms that rely only on Linked Data and showed that our algorithm improves the rate of new recommendations while maintaining a satisfying prediction accuracy.

Focusing on RQ2, Context-Aware Recommender Systems (CARS) aim to provide users with the most useful recommendations for their current situation. However, an exact context obtained from a user could be too specific and may not have enough data for accurate rating prediction [14]. This is a form of data sparsity problem. To represent the user and her context, this thesis presents the Recommender System Context (RSCtx) ontology, which is combined with the Contextual Ontological User Profile (COUP) ontology to generate a new context-aware recommendation approach which can be used with existing recommendation algorithms. We applied RSCtx for context identification and generalization tasks and showed that it is possible

to represent context targeting a user who receives recommendations by combining different dimensions and representing different granularities for each dimension. The evaluation compared our approach with state-of-the-art algorithms and demonstrated how it could significantly improve the prediction accuracy.

Addressing RQ3, the problem was effectively visualizing the recommended resources and their relationships. In fact, a visualization tool for mobile devices which was not limited to a single domain was still lacking. This thesis proposes a visualization framework for DBpedia which is suitable for mobile devices and can be adapted to any dataset in the Web of Data because it is based only on standard Linked Data languages, such as RDF and SPARQL[5] [15] (briefly introduced in Section 2.3.2). The framework has been applied to a mobile application to recommend movies, which was developed in collaboration with Telecom Italia.

In summary, this thesis shows how it is possible to exploit structured data available on the Web to recommend useful resources to users. Linked Data have been successfully applied to recommender systems to provide cross-domain and novel recommendations and address well-known problems such as data sparsity. The challenges previously mentioned originated from the needs of Telecom Italia, which aimed to improve the mobile services offered and to increase the benefit for its users. Various solutions to these problems were applied to specific use cases which were provided by the company through the implementation of prototypes. Additionally, semantic annotation and classification techniques from the state of the art have been applied to some practical use cases provided by the company in order to recommend resources for further information from a given text in the context of social reading and social TV.

The thesis is organized as follows. Chapter 2 provides an overview of recommender systems and the Web of Data, along with a list of challenges. Chapter 3 describes the systematic literature review that we conducted about Linked Data based recommender systems to identify the research issues addressed in this thesis. Chapter 4 introduces Allied, the framework to deploy and evaluate Linked Data based recommendation algorithms in which ReDyAl has been integrated. ReDyAl is presented in Chapter 5, which also describes the evaluation of the algorithms and its application in use cases of Telecom Italia. Chapter 6 proposes RSCtx and shows how it has been applied to identify and generalize context. The chapter also shows

[5] http://www.w3.org/standards/techs/sparql

how RSCtx is combined with COUP to generate a new context-aware recommendation approach and provides the results obtained while evaluating this approach. Chapter 7 describes DBpedia Mobile Explorer, a framework for DBpedia which is suitable for mobile devices and can be adapted to any dataset in the Web of Data. Chapter 8 shows how semantic annotation and classification techniques from the state of the art have been applied to some practical use cases provided by Telecom Italia to recommend resources for further information in social reading and social TV applications. Finally, Chapter 9 summarizes the contributions of this thesis and discusses the open issues.

Chapter 2

Background

2.1 Introduction

The huge amount of data available on the Web, together with the massive publication of user-generated content enabled by social networks and mobile devices generated an information overload: more information is produced than what we can consume.

Automatic filtering tools have become common to assist the user deal with such an enormous amount of information. Recommender Systems (RS) represent a subset of these tools and aim to suggest interesting items to the user. At the same time, the Web is evolving from an information space for sharing textual documents into a medium for publishing structured data. Linked Data is a set of best practices to publish and interlink data on the Web, and it is the base of the Web of Data, an interconnected global dataspace where data providers distribute their content publicly. Information on the Web is no more intended only for humans, in contrast, it is available also for machines.

In this chapter, we firstly review the recommendation problem and the various approaches to address it (Section 2.2), then we introduce the Web of Data basic principles and main features (Section 2.3).

2.2 Recommender Systems

RS are software tools and techniques that provide suggestions of items to be of use to a user [1]. Item is the general term used to indicate what the system recommends. In effect, the suggested items can belong to different categories, e.g. songs, places, news, books, films, events, etc. A recommender system usually focuses on a specific type of item; thus its graphical user interface and the core technique used to provide recommendations are optimized for the particular kind of item addressed.

According to Adomavicius and Tuzhilin [16], the roots of RS can be traced back to the works in cognitive science, approximation theory, information retrieval, forecasting theories, management science, and consumer choice modeling in marketing. Developing RS is a multidisciplinary effort which still requires expertise from these various areas. RS are newer than other classical information management tools such as database and search engines because they emerged as an independent research field since the publication of the first studies on collaborative-filtering (one of the most popular recommendation methods) in the mid-1990s [17–20]. Compared to search systems, recommender systems provide the possibility for users to discover new resources that they may have not initially thought about, without requiring to formulate their needs explicitly [10].

The interest in this area is still high because it is a problem-rich research field and practical applications of RS help users to deal with information overload and provide personalized recommendations, content, and services to them [16]. Highly appreciated websites such as Amazon.com, YouTube, Tripadvisor, Last.fm, and Netflix are examples of these practical applications since RS are their key components.

2.2.1 The Recommendation Problem

The recommendation problem is finding for each user an item which maximizes the utility of the item for the given user. A formal definition is provided by Adomavicius and Tuzhilin [16] and is described in Equation 2.1 .

$$\forall u \in U,\ i^{max,u} = arg\ max_{i \in I}\ f(u,i) \tag{2.1}$$

U is the set of users considered by the recommender system and I the set of items; they can be both extremely large. The utility function $f\colon U \times I \rightarrow R$ represents the

usefulness of an item $i \in I$ for a user $u \in U$, where R is a totally ordered set (e.g. nonnegative numbers within a given range).

The utility of an item is often represented by a rating, which indicates how a particular user liked a given item. For instance, Alice gave the movie *The Green Mile* the rating 4 out of 5. The fundamental problem is that the utility is not defined on the whole $U \times I$ space, but only a subset is available. In fact, only a portion of ratings is known for each user. Thus a recommender system has to assess the utility function from the available data and use it to predict unknown values. Typically the recommendations are provided by selecting for each user the best N items, i.e. the items with the highest utility (top-N recommendations).

RS are information systems which need some input data to generate recommendations. These data are firstly about items and users, but they cannot always be exploited because of the variety of sources and recommendation techniques used. Some methods only require basic data (e.g. ratings), while others are more knowledge dependent, e.g. they may rely on ontological descriptions of users and items, or social relations and activities of users. In any case, three essential elements are involved in RS: users, items, and ratings. The firsts are the actors who are receiving the recommendations; the seconds are the resources to recommend to users; the thirds are the users' preferences (as feedback) and constitute the relations between users and items. These elements are represented differently in the system depending on the recommendation technique used.

The availability of up-to-date users' preferences is often the primary need of RS. Users' feedback can be explicit or implicit according to how it is collected [10]. Ratings are an example of explicit feedback. In this case, users express the opinion about an item through a numerical, ordinal or binary scale. For example, ratings can be respectively given as a number of stars (e.g. from 1 to 5), one value among *strongly agree, agree, neutral, disagree, strongly disagree*, or as *like / dislike*. Anyway, ratings are not the only form of representing utility. For instance, tags such as *too long* or *acting* can also provide some feedback from users [1].

It is important to notice that the primary goal of RS is predicting ratings according to our definition of the recommendation problem. In the literature, this formulation is referred to as rating prediction task [10]. Nevertheless, often RS have to provide the user with a ranked list of recommendations and, in many commercial systems, the *best bet* recommendations are shown, but the predicted rating values are not [21].

This is known as top-*N* recommendation task [10, 21], ranking task [22], or item recommendation task [23] since the emphasis shifted from predicting ratings to ordering items according to the user's preferences.

2.2.2 Recommendation Techniques

Various recommendation approaches have been proposed. They differ in the assessment of the utility function and in the data exploited. Typically RS are classified according to the following main categories: content-based, collaborative filtering, knowledge-based, and hybrid [24].[1] Furthermore, Adomavicius and Tuzhilin [16] also distinguish between heuristic-based and model-based on based on the techniques used for the rating estimation. Since these two categorization are orthogonal, each type of RS can be further classified as heuristic-based or model-based. In the following, we introduce the essential features of each type of RS and we summarize their strengths and weaknesses.

Content-based RS make suggestions based on the ratings that users gave to items and the content of the items (e.g. extracted keywords, title, pixels, disk space, etc.) [25]. Basically, these systems recommend items which are similar to the one that a given user liked in the past. For example, in a movie recommendation scenario, Alice might like *The Shawshank Redemption* because she liked *The Green Mile*. The similarity is based on the features which describe the item, e.g. the genre of a book or the optical zoom of a camera. These features can be extracted from unstructured or semi-structured data through text mining techniques or obtained from structured data (e.g. relational databases).

Heuristic-based RS represent both items and users using Information Retrieval (IR) techniques (e.g. vectors of terms) and compute the similarity between their representations. The user profile is a vector of terms built from the analysis of the items liked by the user. The Vector Space Model (VSM) [26] is a heuristic approach widely used which models items and user profiles as weighted vectors typically computed with the Term Frequency-Inverse Document Frequency (TF-IDF) formula [26]. This method is often combined

[1] A deep review of the recommendation approaches is out of the scope of this thesis. The reader is encouraged to refer to the book of Ricci et al. [1] or the survey of Adomavicius and Tuzhilin [16] for an exhaustive discussion. Burke [24] provided a widely used taxonomy which provide an overview of the several types of RS.

	The Green Mile	The Matrix	Slevin	The Last Samurai
Alice	5	?	2	?
Bob	?	3	5	1
John	4	?	?	5

Table 2.1 A user-item matrix in a movie recommendation scenario.

with the cosine similarity to assess the similarity between items and user profiles in order to recommend the items most similar to the user profile.

Model-based approaches use Machine Learning techniques to generate a model of the user's preferences by analyzing the features of items that the user have rated [10]. For each user, a regression or classification model is learned from a collection of items for which ratings are available. The training set consists of item feature vectors labeled with ratings. This model is used for estimating the unknown ratings.

Lops et al [25] outlined the main pros and cons of this kind of system. The advantages are user independence, transparency, and new item. The former is the need of only ratings of the user considered. The second allows the system to explain the recommendation provided by using the features of the items recommended. The latter is the ability of recommending items which have not yet been rated. However, content-based RS have some limitations. They cannot provide recommendations which differ from the items already known by the user. This issue is known as overspecialization. Additionally, these systems relies on features extracted from the item content. Thus the quality of the recommendation provided depends on the availability and the quality of the features. This problem is called limited content analysis. Finally, a new user is an issue since enough ratings have to be collected before a recommender system can provide accurate recommendations.

Collaborative-filtering RS recommend items to a user taking into account ratings that users with similar preferences gave to these items [27]. The similarity of the users is based on the similarity of their past ratings. Collaborative-filtering is the most implemented technique [1]. Its primary advantage is needing only ratings, which can be stored and processed efficiently being relatively simple information. Thus, a second advantage is scalability.

Usually, these systems rely on a user-item matrix, which is a table representing a user in each row, an item in each column and a rating given by a user to an item in each cell. This matrix is typically sparse because users rate few items. Table 2.1 shows an exemplifying matrix for a movie recommender system. For instance, Alice might like *The Last Samurai* because John liked it and John and Alice both gave a high rating to *The Green Mile*.

Heuristic-based methods (also called memory-based [28]) rely on the k-nearest neighborhood algorithm (kNN). It allows the system to predict missing rating by aggregating the ratings of the closest neighborhood. Heuristics can be user-based or item-based. The former recommends an item to a user through a linear combination of neighbor's ratings which weights the similarity between these neighbors and the user. An example of this approach is given one of the first RS [19], which uses the cosine similarity to estimate the similarity between items in order to infer the item similarity from the user's ratings. In this way, the system can suggest items similar to those the user has already liked. Lately, model-based techniques have emerged because they are more accurate than heuristic-based approaches [29]. The most popular model-based methods are based on matrix factorization. They reduce the user-item matrix to map users and items in a joint lower dimensional latent factor space [30].

The main weakness of collaborative-filtering RS is that they need to know a certain amount of ratings before providing proper recommendations. Cold-start and data sparsity are two problems related to this need. The former occurs when the user has not rated enough items to allow the system to compute the similarity with other users [31], while the latter arises because users usually rate a small portion of the available items [32]. With sparse ratings, two users or items are unlikely to have common ratings. Thus ratings are predicted with a low number of neighbors. Moreover, recommendations may be biased because the similarity weights may be computed using only a small number of ratings. For this reason, new users and new items are also problematic. In fact, the system cannot calculate reliable similarities with other users until the user has not rated a sufficient number of items, while a new item cannot be recommended before being rated.

Knowledge-based RS infer and analyze similarities between user requirements and features of items described in a knowledge base that models users and items

according to a particular application domain [33]. Two types of knowledge base RS which are worth to mention are case-based and constraint-based. They are similar in term of knowledge because they both collect user requirements, can restore inconsistent requirements when no solution is found and they can explain the recommendations generated, but they differ in the computation of recommendations [1]. While content-based and collaborative-filtering RS suit to scenarios with users interacting with the system over time, knowledge-based RS do not require an interaction history. Their primary weakness is the high cost to model and maintain the knowledge exploited. Typically, knowledge bases require constant updates due to the changes of item features and user requirements.

With the evolution of the Web towards a global space of connected and structured data, a new kind of knowledge-based RS has emerged known as Linked Data based RS. These systems suggest items taking into account the knowledge of datasets published under the Linked Data principles. The works belonging to this category are reviewed in Chapter 3.

Method	Description
Weighted	The scores of several recommendation techniques are combined together to produce a single recommendation.
Switching	The system switches between recommendation techniques depending on the current situation.
Mixed	Recommendations from several different RS are presented at the same time.
Feature combination	Features from different recommendation data sources are combined into a single recommendation algorithm.
Cascade	One recommender refines the recommendations given by another.
Feature augmentation	The output of one technique is used as an input feature of another.
Meta-level	The model learned by one recommender is used as input of another.

Table 2.2 Hybridization methods [34]

Hybrid RS combine one or more of the techniques previously mentioned to improve recommendations. They aim at compensating the weakness of a technique with the strength of another. Burke [34] proposed a classification of possible hybrid techniques which is summarized in Table 2.2. A widely adopted hybridization is the combination of collaborative-filtering and content-based approaches to

Method	Advantages	Disadvantages
Content-based	User independence, transparency, new item	Overspecialization, limited content analysis, new user
Collaborative-filtering	They require only user ratings, scalability	Cold-start, data sparsity, new user, new item
Knowledge-based	They do not require any interaction history	High cost to model and maintain the knowledge-base
Hybrid	Mitigate the drawback of one technique with the strength of another	Possibly poor performance

Table 2.3 Comparison of the recommendation techniques described in this section.

mitigate cold-start and data sparsity problems. However, due to their inner complexity these systems may have poor performance.

Context-Aware Recommender Systems (CARS) are a particular category of hybrid RS, which exploits contextual information to provide more useful recommendations. For example, in a temporal context, vacation recommendations in winter should be very different from those offered in summer, or a restaurant recommendation for a Saturday evening with your friends should be distinct from that suggested for a workday lunch with co-workers [1]. Section 6.2 provides a more detailed description of these systems.

The pros and cons of each technique are summarized in Table 2.3.

2.3 The Web of Data

The Web has evolved from an information space for sharing textual documents into a medium for publishing structured data. The Linked Data[2] initiative encourages the publication and interlinking of data on the Web, generating the Web of Data, a global dataspace where content is public and interconnected [35]. In this section, we firstly introduce the needs which the Web of Data aims to address, and then we describe the Linked Data principles and technology stack.

[2] http://linkeddata.org

2.3.1 Beyond Data Silos

The Web of Data goes beyond the traditional Web made up of HTML pages which can be read by humans and with hyperlinks manually created. The idea of extending the capabilities of the Web to publish structured data on it exists from its creation. Tim Berners-Lee[3] [36] highlighted the need of introducing semantics into the Web in order to achieve this idea, which later became known as Semantic Web.[4] While the Semantic Web is the goal, Linked Data provides the means to make it reality [37]. It refers to a set of best practices for publishing and connecting structured data on the Web to increase the number of data providers and consequently accomplish the goals of the Web of Data. In this way, Linked Data makes it possible to semantically interlink and connect different resources at the data level regardless of their structure, location, etc.

To achieve the Semantic Web, we need to deal with implementing such Web of Data, i.e. publishing data so that reuse is encouraged and fostering data integration from many different sources. Sharing and reuse are possible only if data is structured. Solutions as microformats or Web APIs are either too domain specific or need ad-hoc consumption techniques. Furthermore, data integration and discovery are possible only when a shared data model is adopted across systems, along with standard data schemas. For instance, consider the "data silo" problem on the Web, i.e. set of APIs expose single datasets, but no external datasets connections are provided, thus losing an appealing feature of the Web, the hyperlinks between entities. Thus, the birth of Linked Data was a fundamental step towards the Web of Data, since they encourage the generation of a network of interconnected datastores on the Web, going beyond data silos [35]. Figure 2.1 depicts the Web of Data as the Linked Data Cloud diagram, the Web of Data includes but is not limited to the datastores represented. The figure shows a number of interconnected datasets which belong to different application domains, such as academic publications, social network, government, and life science. The estimated size of the Web of Data is numerous billion of data statements (known as RDF triples, which are introduced in Section 2.3.2) [35].

[3] http://www.w3.org/Talks/WWW94Tim
[4] http://www.w3.org/standards/semanticweb/
[5] http://lod-cloud.net/
[6] http://www.w3.org/2007/03/layerCake.png

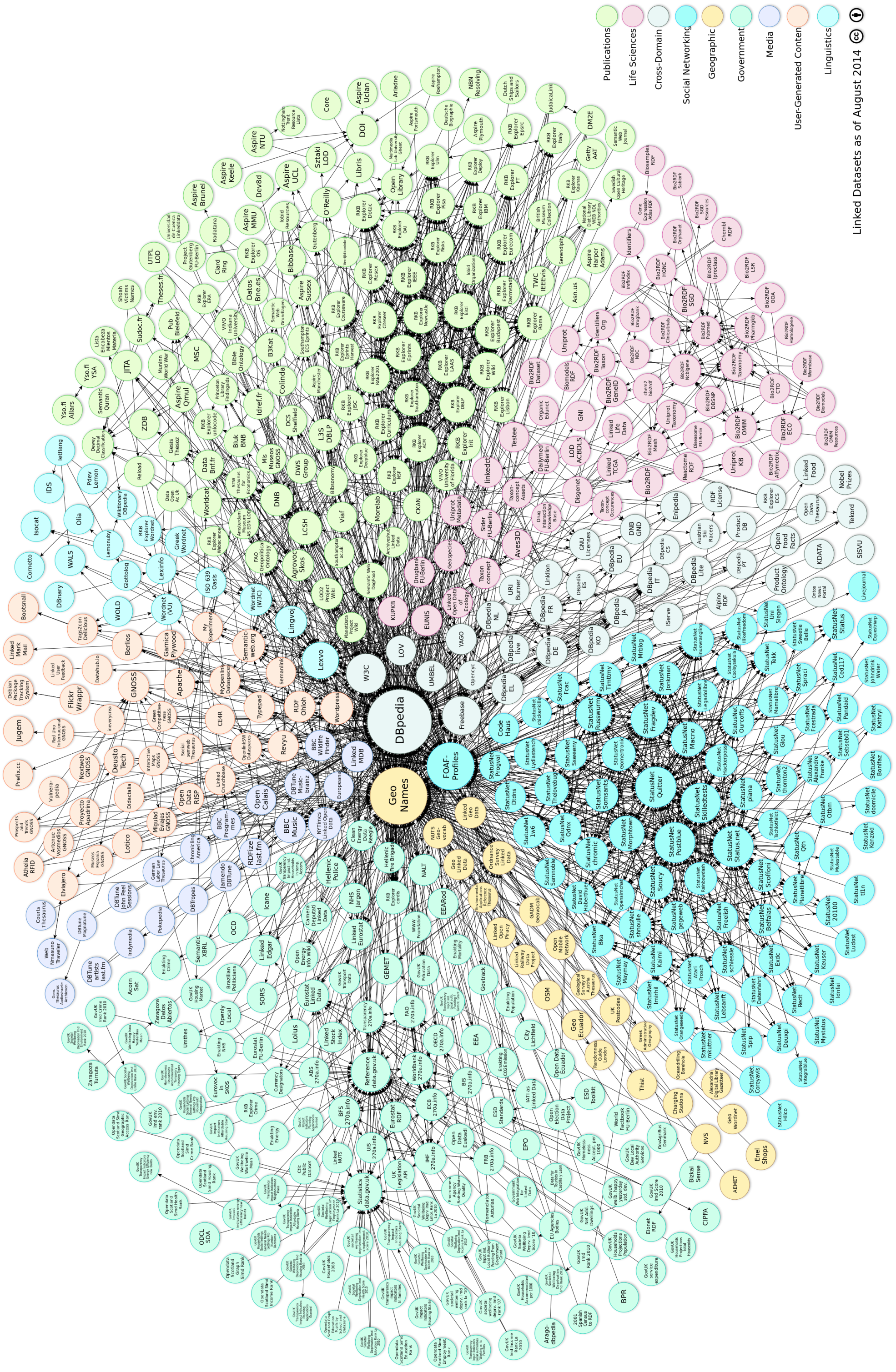


Fig. 2.1 The Linked Data Cloud as in 2014.[5] Each circle is a dataset published and interlinked on the Web following the Linked Data principles. The size of the circle represents the size of the datasets, while colors indicate the application domains. Arrows means at least 50 links with external datasets.

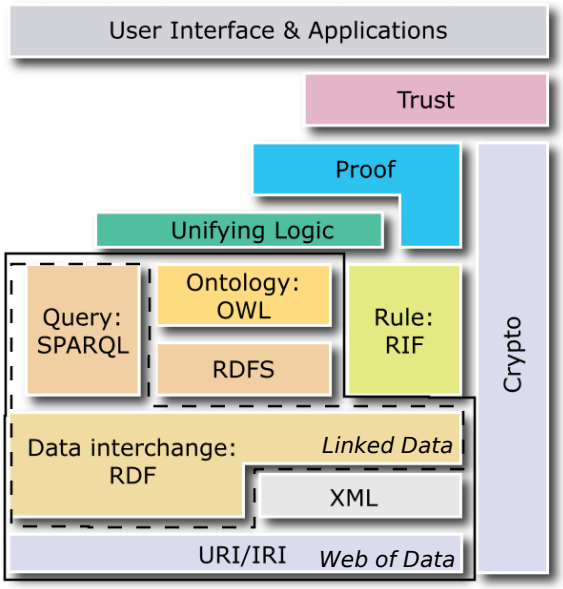


Fig. 2.2 The Semantic Web stack.[6] The continuous line includes the technologies used in the Web of Data, while the dotted line incorporates Linked Data technologies.

2.3.2 Technology Stack and Linked Data Principles

As shown in Figure 2.2, Linked Data relies on a number of technologies which are a subset of the Web of Data. The latter is in turn a subset of the Semantic Web.

HTTP Universal Resource Identifiers (URI)[7] generalize Universal Resource Locators (URL). The first identifies any kind of resource (such as individuals, real-world objects, etc.), while the latter refers to web pages only.

The Resource Description Framework (RDF)[8] [11] structures information as labeled graphs. The graphs are represented by means of triples, which are statements made up of three elements: a subject, a predicate (also called property) and an object. The first and the third element are node in the graph, while the second is an arc directed from the subject to the object. URIs are the identifiers used in RDF. For example, Figure 2.3 depicts shows a simple RDF representation of *The Matrix*. While at conceptual level RDF data are graphs, at machine level there is a number of serializations for these data and each one has its own format. The most used are

[7] http://www.ietf.org/rfc/rfc2616.txt
[8] http://www.w3.org/standards/techs/rdf

Turtle [38] and RDF/XML [39] (XML based), and JSON-LD[9] (JSON for Linked Data, which extends JSON).

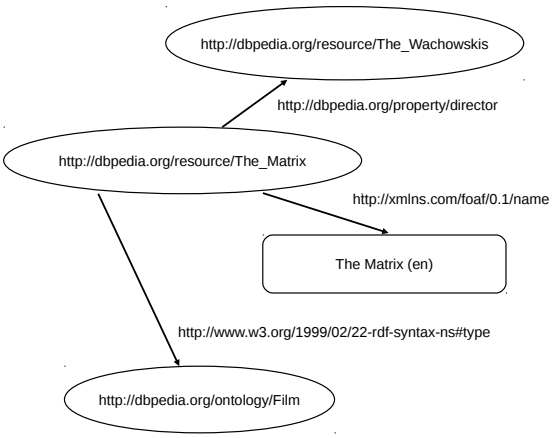


Fig. 2.3 An exemplifying graph representation of *The Matrix*.

Another important point in the interconnecting data is to integrate them to foster the reuse. So it is possible to avoid describing the same subject twice, and for this reason, data are organized through vocabularies and ontologies. They define the concepts and relationships (also referred to as terms) used to describe and represent an area of concern. They are used to classify the terms that can be used in a particular application, characterize possible relationships, and define possible constraints on using those terms.[10] There is no clear division between vocabularies and ontologies. Usually ontology indicates a more complex, and possibly quite formal collection of terms, while in a vocabulary such strict formalism is not necessary. Examples of vocabularies are Dublin Core, and Friend Of A Friend (FOAF), respectively to describe digital documents and people or information on the Web. RDFS [40] and OWL[11] are languages to define Web of Data vocabularies. While RDFS allows specifying classes and properties of resources together with domain and range of relations, OWL supports more advanced features and increases the expressiveness.

SPARQL[12] allows consuming data. In fact it is the language to query and update RDF data and it is similar to SQL in the syntax and in the vocabulary, although it

[9] http://json-ld.org/
[10] https://www.w3.org/standards/semanticweb/ontology
[11] http://www.w3.org/standards/techs/owl
[12] http://www.w3.org/standards/techs/sparql

is executed over graphs instead of tables. An example of a query that selects the director of the movie *The Matrix* is provided in Listing 2.1.

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT ?director WHERE {
  dbpedia:The_Matrix dbpedia-owl:director ?director .
}
```

Listing 2.1 A SPARQL query to retrieve the director of *The Matrix*.

Tim Berners-Lee relied on the aforementioned standards to define the best practices for publishing data on the Web, known as Linked Data principles.[13] These principles are:

1. use URIs as names for things;
2. use HTTP URIs so that people can look up those names;
3. when someone looks up a URI, provide useful information, using the standards (RDF, SPARQL);
4. include links to other URIs so that they can discover more things.

The first principle states that URIs must be used to name "things". URIs identify real world objects, abstract concepts , and relationships between objects or resources. The second principle recommends combining the HTTP protocol and URIs to retrieve the desired resource. The third principle proposes the adoption of the RDF data model. The fourth principle highlights the importance of linking resources to others, as done with hyperlinks in HTML pages. Links connecting resources are typed (using RDF); thus unlimited types of relationships might be created. Since RDF links may interconnect resources hosted in different datasets, implementing these principles results in a globally interconnected data space: the Web of Data.

[13] http://www.w3.org/DesignIssues/LinkedData.html

Chapter 3

Linked Data Based Recommender Systems

3.1 Introduction

Nowadays, RS are increasingly common in many application domains, as they use analytic technologies to suggest different items or topics that can be interesting to an end user. However, one of the biggest challenges in these systems is to generate recommendations from the large amount of heterogeneous data that can be extracted from the items. Accordingly, some RS have evolved to exploit the knowledge associated to the relationships between data of items and data obtained from different existing sources [1]. This evolution has been possible thanks to the rise of structured data published on the Web, such as Linked Data.

This chapter summarizes the state of the art of RS that make use of the structured data published as Linked Data on the Web. We undertook a systematic literature review, which is a form of secondary study that uses a well-defined methodology to identify, analyze and interpret all available evidence related to specific research questions in a way that is unbiased and (to a degree) repeatable [41, 42]. We considered the most relevant problems that RS are intended to solve, the way in which studies addressed these problems using Linked Data, their contributions, application domains and the evaluation techniques that have been applied to assess their recommendations. Analyzing these aspects, we deduced current limitations and possible directions of future research. Unlike other works reporting the state of the art

in RS [16, 25, 43, 44] our systematic literature review is the first to extensively study RS that obtain information from Linked Data in order to generate recommendations. Some approaches where mentioned in the survey of Marie and Gandon [45], but it focused on a different topic.

The remainder of this chapter is structured as follows: Section 3.2 summarizes the methodology and defines objectives and research questions. Section 3.3 outlines the results of the review organized according to each research question defined in section 3.2. Section 3.4 discusses the results as well as the limitations of our systematic literature review. Section 3.5 contains the conclusions and future work. We list the selected and excluded papers in Appendix A.

3.2 Research Methodology

This chapter studies the state of the art in Linked Data based RS. It follows the guidelines set out by Kitchenham and Charters [42] for systematic literature reviews in software engineering. These guidelines provide a verifiable method of summarizing existing approaches as well as identifying challenges and future directions in the current research. Figure 3.1 presents the protocol for our systematic literature review. The protocol's purpose is defining the steps to conduct the review.

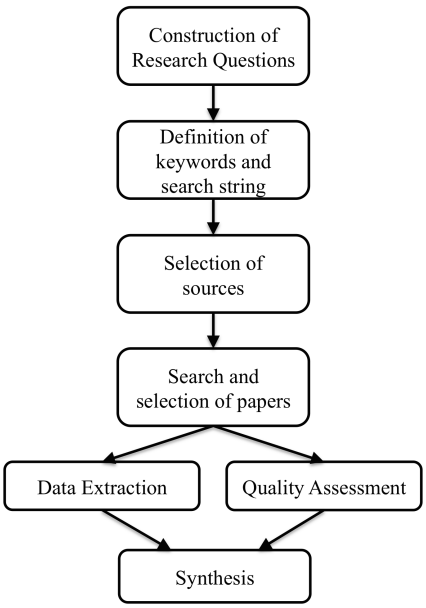


Fig. 3.1 The process of our systematic literature review.

3.2.1 Research Questions, Search String, and Sources

The goal of our systematic literature review is to understand how the implicit knowledge, stored in Linked Data datasets and represented as concepts and relations between them, can be exploited to make recommendations. Accordingly, we have defined the following research questions:

RQ1 What studies present RS based on Linked Data?

RQ2 What challenges and problems have been faced by researchers in this area?

RQ3 What contributions have already been proposed (e.g. algorithms, frameworks, engines)?

RQ4 How is Linked Data used to provide recommendations?

RQ5 What application domains have been considered?

RQ6 What criteria and techniques are used for evaluation?

RQ7 Which directions are the most promising for future research?

Afterward, a preliminary set of keywords was defined: {*Linked Data, Recommender system*}. This set was then extended by searching for synonyms in order to obtain the final set of keywords used to define a search string. The search string is the query to look for papers in a set of online digital libraries and the one exploited in our case is showed in Listing 3.1. To decide which synonyms we needed to include in the search string, we relied on a set of relevant papers we were familiar with. This set is available in Appendix A. The author and a colleague independently selected a subset of the keywords used in these papers and merged them. We solved disagreements through discussion.

```
("semantic web" OR "linked data" OR "web of data" OR
"linked open data") AND (recommendation OR
"recommender system" OR "recommendation system" OR
"semantic recommendation" OR "semantic recommender").
```

Listing 3.1 The search string used to look for papers in digital libraries.

Source	URL
IEEExplore	http://ieeexplore.ieee.org
SpringerLink	http://link.springer.com
Scopus	http://www.scopus.com
ACM Digital Library	http://dl.acm.org
Science Direct	http://www.sciencedirect.com
ISI Web of Knowledge	http://apps.webofknowledge.com
Wiley Online Library	http://onlinelibrary.wiley.com

Table 3.1 The sources selected for our search process.

Furthermore, we selected seven scientific digital libraries that represent primary sources for computer science research publications as can be seen in Table 3.1. Kitchenham and Brereton [46] recommend relying on IEEE and ACM to assure a good coverage of relevant journals and conferences and at least two general indexing systems, which were Scopus and ISI Web of Knowledge in our case. We also included Springer and Science Direct because they publish important journals and conferences related to Semantic Web, Linked Data, and RS. Other sources like DBLP, CiteSeer, and Google Scholar were not considered as they mainly index data from the primary sources. We excluded these digital libraries because we believe the risk of missing some contribution is low, while the time and effort to search them are high due to the large number of results returned.

3.2.2 Search and Selection

The studies selected for this systematic literature review were identified from the selected sources during March 2014. In Table 3.2, a set of inclusion/exclusion criteria were defined in order to determine whether or not a study should be included.

3.2.3 Quality assessment, Data Extraction and Synthesis

The goal of quality assessment is excluding low quality papers because they results may be biased. We have defined a set of quality criteria that are listed in the checklist provided in Table 3.3. Quality for each question is typically scored with values 1,

[1] A deep discussion of Linked Data based exploratory search systems is out of the scope of this thesis. The reader may refer to the survey of Marie and Gandon [45].

Inclusion Criteria
Papers presenting RS using Linked Data to provide recommendations.
Papers addressing exploratory search systems using Linked Data.[1]Exploratory search refers to cognitive consuming search tasks such as learning or topic investigation [47]. Exploratory search systems also recommend relevant topics or concepts, although the key difference with respect to RS is that they still require an input query (commonly a set of keywords).
Papers from conferences and journals.
Papers published from 2004 to 2014. Linked Data is a relatively new technology, therefore RS approaches exploiting it are also recent.
Only papers that are written in English.
Short and workshop papers which fulfill the above criteria: we had no reason to believe that they would fail to provide sufficient levels of detail about their studies.
Exclusion Criteria
Papers not addressing RS neither exploratory search systems.
Papers addressing RS or exploratory search systems that do not exploit Linked Data to produce recommendations.
Papers addressing similarity measures but not RS. Similarity is a broader topic than RS.
Papers which use Semantic Web techniques (e.g. rule-based or ontology-based reasoning) and knowledge bases but not Linked Data. Linked Data based RS have specific needs to integrate Linked Data, as explained in Chapter 1.
Abstracts or slides of presentations because of the lack of information.
Grey literature. We do not think that technical reports, unpublished studies, and Ph.D. thesis would add much more information with respect to journal and conference papers.

Table 3.2 Inclusion and exclusion criteria.

0.5, and 0, in order to represent the answers *'yes'*, *'partly'* and *'no'*. The author and another colleague evaluated the studies selected using this checklist. To do this, the total set of selected papers was split into two disjoint subsets and each researcher selected only one of these subsets to evaluate the papers. After this evaluation,

Question	Score
Q1. Did the study clearly describe the challenges and problems that is addressing?	yes / partly / no (1 / 0.5 / 0)
Q2. Did the study review the related work for the problem?	yes / partly / no (1 / 0.5 / 0)
Q3. Did the study discuss related issues, and compare with the alternatives?	yes / partly / no (1 / 0.5 / 0)
Q4. Did the study recommend the further continuous research?	yes / partly / no (1 / 0.5 / 0)
Did the study describe the components or architecture of the proposed recommender system?	yes / partly / no (1 / 0.5 / 0)
Q5. Did the study describe the components or architecture of the proposed recommender system?	yes / partly / no (1 / 0.5 / 0)
Q6. Did the study provide empirical results?	- The study provided an implementation of its work with an empirical evaluation and it was used in real applications, e.g. by other services (1) - The study provided an implementation of its work and an empirical evaluation but was not referred or used in other studies/applications (0.75) - The study provided an implementation only (0.5) - The study did not provide any implementation but it was referred by other works as a base on which start (0.25) - The study did not provide any implementation and was not referred by other works (0)
Q7. Did the study provide a clear description of the context in which the research was carried out?	yes / partly / no (1 / 0.5 / 0)
Q8. Did the study present a clear statement of findings?	yes / partly / no (1 / 0.5 / 0)

Table 3.3 Quality assessment checklist.

cross-checking of the assessment was done on arbitrary studies (about 30% of the selected papers) by a third colleague. Agreement on differences was reached by discussion. Finally, for each study, we computed a quality score as the average of the scores of the individual question and we decided to exclude papers with a score lower than 0.5.

Data extraction was done in parallel with the quality assessment. We split the set of included studies into two disjoint subsets. The author and another colleague performed the task on a subset, then a third colleague cross-checked a random sample of 30% of the studies. The data extracted are presented in Table 3.4.

The synthesis step is based on the methodology for thematic synthesis described by Cruzes and Dybå [48]. This methodology define codes as descriptive labels applied to segments of text from each study. We defined an initial set of codes based on research questions and, subsequently, we performed a second coding with more precise codes, which were closer to the content of selected papers. The coding was performed by the author and another colleague: each of them addressed a subset of the papers as for data extraction and quality assessment since it was done in parallel with them. Then, a third colleague performed again the coding on a random sample of 30% of the papers for cross checking, afterward, disagreements were solved by discussion. The codes used are showed in Appendix A.4.

3.3 Results

This section summarizes the relevant information found in the studies selected in order to answer the proposed research questions. A further discussion and analysis of these results are addressed in Section 3.4.

3.3.1 Included Studies

RQ1 regards the studies that present RS based on Linked Data. We retrieved 69 papers to include in the systematic literature review, corresponding to 52 unique primary studies (a study is a unique research work that can include one or more papers). These studies were published in conferences, workshops and journals between 2004 and 2014. The criteria for deciding the most significant paper for each

Data Field	Description	Research Question
ID	-	-
Title	-	-
Authors	-	-
Year of publication	-	-
Year of conference	-	-
Volume	-	-
Issue	-	-
Location	-	-
Proceeding title	-	-
ISBN	-	-
Publisher	-	-
Examiner	Name of person who performed data extraction	-
Publication source	-	-
Context	Environment in which study was conducted: industry, academic, government	-
Population	Study participants: students, academics, practitioners, etc.	-
Aims	Goals of the study (in our opinion when not clearly reported by authors)	-
Research problem	-	RQ2
Application domain	-	RQ5
Contributions	-	RQ3
Criteria and techniques for evaluation	-	RQ6
Findings	-	-
Limitations	-	RQ7
Future work	-	RQ7
Notes	-	-
Other Information	-	-

Table 3.4 Data extraction form.

study were completeness and publication year. The final set of selected papers and corresponding studies can be found in Appendix A.

Concerning the quality assessment, the quality score was higher than 0.5 for all papers i.e. rather good according to the quality criteria defined in Section 3.2.3. Thus, we did not exclude any papers because of its quality. In fact, the goal of quality assessment is to avoid to include low quality studies since their results could be biased. It is not a result of the systematic review. As shown in Figure 3.2, Q7 (*Did*

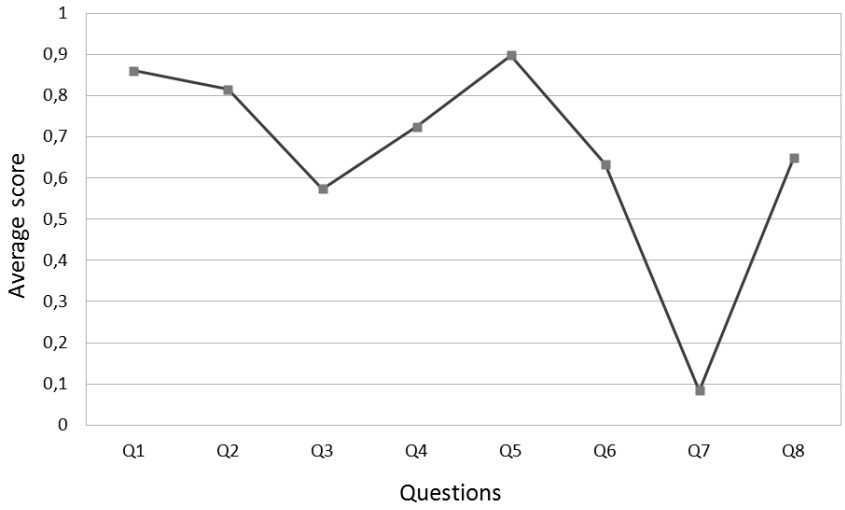


Fig. 3.2 Average quality score per question.

the study provide a clear description of the context in which the research was carried out?) was not applicable. In fact, we often answer negatively to this question and the resulting average score is lower than 0.5. On the contrary, the other questions are acceptable as their average score is close or above 0.6.

3.3.2 Research Problems

In order to address RQ2, we summarize the main problems involved in the studies considered that regard the production of accurate recommendations. Table 3.5 lists these problems according to the number of studies in which they occurred. The number of studies represents the occurrence of each problem in the studies selected, which may be addressed in more than one study. The same applies for the rest of the results reported in this section.

In the following we describe each item of Table 3.5:

Lack of semantic information It was the most frequent problem in the studies selected and it concerns the need for exploiting the rich semantics of information about items. Possible causes of this problem are: data about items are unstructured; a categorization of the items is needed; it is necessary to find relationships to link items; social information is lacking; it is necessary to acquire content descriptive metadata; similarity measures that take into account semantic information are needed.

Problems	Number of studies
Lack of semantic information	13
Complexity of information about items	12
User dependency	8
Cold-start	6
Data quality	6
Computational complexity	5
Data sparsity	5
Domain dependency or specific and limited domain	4
Other problems	2

Table 3.5 Distribution of the studies selected according to the problems that they addressed.

Complexity of information about items It is related to the complexity of information due to noisy metadata about features of items. Other causes for this problem are semantic heterogeneity and distribution of resources. The latter can impact on maintenance of the knowledge bases and can also decrease the accuracy of recommendations.

User dependency In a number of cases RS requires users to perform manual operations to acquire information about their profiles and interests. Such operations can be user feedback, ratings, filtering, attaching content-descriptive metadata and semantic annotation of items.

Cold-start It is a well-known problem found mainly on collaborative-filtering RS. Cold-start is a situation in which there are not enough ratings for items in order to generate recommendations, for instance in the case of a new user.

Data quality This problem occurs when the knowledge base used to acquire information for providing recommendations is not reliable. Problems affecting data quality can range from poor reliability (e.g. wrong links between concepts, or incorrect representations) to poor quality of recommended items.

Computational complexity It is related to the high computational demand that RS require to produce recommendations due to the large amount of data about items.

Data sparsity This is related to the lack of information about users or items and generates a low density of significant data or connections.

Contribution	Number of study
Algorithms	27
Similarity measures	12
Ontologies	8
Information aggregation or enrichment	8
Others	16

Table 3.6 Distribution of the studies selected according to the contributions provided.

Domain dependency It occurs when recommendations are only useful for items in a specific and limited domain without taking into account data that can be obtained from other related domains.

Other problems They include the need for recommending relevant and yet unknown items and the overspecialization of RS.

3.3.3 Contributions

In order to address RQ3, we classified the contributions provided by each study. Table 3.6 shows the different kinds of contributions and the number of studies in which they occurred (each study possibly reports more than one contribution).

The two main contributions are the definition or extension of a similarity measure and the definition or extension of an ontology, accounting for 12 and 8 studies respectively. Algorithms are also addressed by 27 studies in total. Finally, information aggregation or enrichment and various other contributions account for 8 and 16 studies, respectively. In the following, we describe each item of Table 3.6:

Algorithms Most of the studies selected proposed new algorithms or extensions of algorithms existing in the literature. In particular, four categories emerged: (*i*) defining a new algorithm, (*ii*) adapting an algorithm to Linked Data, (*iii*) combining algorithms to obtain a new hybrid algorithm, and (*iv*) extending an existing algorithm. The definition of a new algorithm was the most frequent with 15 studies, while the adaptation of an algorithm to Linked Data, the combination of algorithms to obtain a new hybrid algorithm and the extension to an algorithm each account for 4 studies. Furthermore, we can group algorithms into two classes:

- Graph-based algorithms, which compute relevance scores for items represented as nodes in a graph. A number of algorithms in this category are: (*i*) the weight spreading activation algorithm, which propagates the initial score of a source node through its weighted edges; (*ii*) algorithms that update the scores of its linked nodes; (*iii*) algorithms that explore concepts and relations defined in an RDF graph; (*iv*) topic based algorithms, which find similar items belonging to the same categories of an initial concept, and (*v*) path-based algorithms to find semantic paths between documents in the RDF graph.
- Algorithms to produce recommendations based on statistical information techniques applied to Linked Data such as Support Vector Machine (SVM), Latent Dirichlet Allocation (LDA), Random Indexing (RI) and scaling methods. SVM analyzes and recognizes patterns in RDF triples; LDA is based on the co-occurrence of terms; RI uses distributional statistics to generate high-dimensional vector spaces; scaling methods take into account the probability that an item could be selected based on its popularity (the number of entities directly connected to the node). In addition, some algorithms define item-user matrices to compute semantic similarity based on path-lengths.

Similarity measures The studies selected applied a variety of similarity measures. These include pairwise cosine function for vector similarity computation between items, feature-based similarity to evaluate semantic distance on different datasets, rating-based similarity to compute the popularity of items among users, semantic relatedness defined by vocabulary meta-descriptions, content similarity that exploits lexical features, expressivity closeness based on the language constructs adopted, distributional relatedness derived from vocabulary usage, and topic-based similarity that captures the relatedness between items based on the categories they belong to.

Ontologies A number of studies proposed ontologies to assist or improve the recommendation process. New ontologies were proposed to facilitate the integration of datasets from a number of domains in order to make RS more flexible to changes, while a combination of existing ontologies described different types of entities such as users and items. Furthermore, it was found that reusing existing ontologies or vocabularies enables interoperability. Ontolo-

gies are also used to represent semantic distances, their explanations, user preferences and item contents. A number of ontologies which are used in studies selected for these purposes are FOAF (Friend Of A Friend),[2] SIOC (Semantically-Interlinked Online Communities),[3] Resource List Ontology,[4] and the Bibliographic Ontology.[5]

Information aggregation or enrichment This refers to the contributions about the aggregation of data into item collections and enrichment of existing ontologies or vocabularies. For example, this is useful to obtain descriptive information about items and find entities in datasets in order to infer links between them. One contribution of this type is the aggregation of information from a specific domain when items have to be enriched with the knowledge contained only on specialized datasets, another is enriching databases of RS with shared vocabularies.

Others Other contributions include the integration of other techniques such as opinion aggregators, exploitation of trust in web-based social networks to create predictive RS and the use of social-based algorithms to improve the performance of the RS.

3.3.4 Use of Linked Data

Another interesting aspect that we studied was the use of Linked Data in RS, as underlined by RQ4. We classified the studies selected according to the way they used Linked Data to produce recommendations and grouped them into:

Linked Data driven RS that rely on the knowledge represented as Linked Data to provide recommendations. For example, RS that calculate a semantic similarity based on diverse relationships that can be found between concepts of Linked Data datasets and are related to features or descriptions of items. Such relationships can be paths, links or shared topics among a set of items. This category can also include RS that use other techniques applied on data

[2] http://www.foaf-project.org/
[3] http://rdfs.org/sioc/spec/
[4] http://vocab.org/resourcelist/
[5] http://bibliontology.com/

obtained from Linked Data datasets, for example, weight spreading activation, Vector Space Model (VSM), SVM, LDA, and RI.

Hybrid RS that exploit Linked Data to perform some operations that can be used or not to provide recommendations. This means that Hybrid RS include Linked Data driven RS, which use recommendation techniques that rely on Linked Data, and RS that use Linked Data in other operations which can be preliminary to the recommendation process (e.g. aggregating more information from other datasets, describing user profiles or annotating raw data in order to extract information to be integrated and used to recommend).

Representation only RS in this category exploit the RDF format to represent data and use at least one vocabulary or ontology to express the underlying semantics. However, no information is extracted from other dataset and Linked Data are not used to provide recommendations. An example is an RS that represents the information about the users according to FOAF vocabulary but does not exploit Linked Data for other operations.

Exploratory search These systems are not RS, but their main duty is to assist users to explore knowledge and suggest topics or concepts relevant to an initial one. Exploratory search systems and RS use Linked Data in a similar way, although the key difference is that exploratory search systems require an explicit input query (commonly a set of keywords). Additionally, users in these systems are not only interested in finding items, but also in learning, discovering and understanding novel knowledge on complex or unknown topics [49].

Each study may be assigned to more than one category, i.e. it can be both Linked Data driven and hybrid, or both exploratory search and Linked Data driven. Studies which are representation only cannot belong to other categories.

Table 3.7 shows that most of the studies considered are Linked Data driven, and roughly 60% of them are also hybrid. Only 20% of hybrid studies were hybrid only, while the rest are also Linked Data driven. Moreover, 10 studies are representation only and just 4 exploratory search systems were included in the systematic literature review. All of the exploratory search studies are also Linked Data driven. This finding is consistent with the focus of the systematic literature review, which is on RS using Linked Data. It is worth noting that exploratory search is a broader topic: we only consider the exploratory systems that recommend concepts to users.

Category	Number of studies
Linked Data driven	37
Hybrid	29
Hybrid and Linked Data driven	21
Linked Data driven only	13
Representation only	10
Hybrid only	6
Exploratory search	4
Exploratory search and Linked Data driven	4
Exploratory search only	0

Table 3.7 Distribution of the studies selected according to their use of Linked Data.

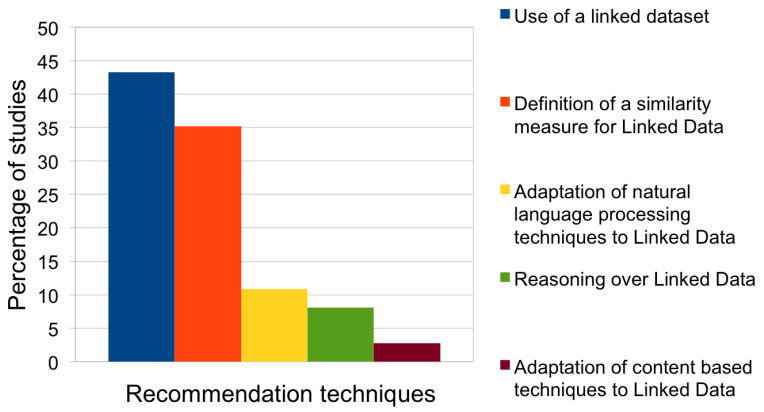


Fig. 3.3 Distribution of the Linked Data driven studies according to the recommendation techniques that they exploit. Percentages refer to the total number of these studies.

The two most interesting categories are Linked Data driven and hybrid. Figure 3.3 shows the different techniques used by the studies in the first category to provide recommendations. The majority of them rely on datasets or on a similarity measure (respectively about 43% and 35%), while the remaining 22% adapt natural language processing or content based techniques or exploit reasoning. Instead, Figure 3.4 illustrates the techniques that hybrid studies use together with Linked Data to provide recommendations. Most of them are natural language processing or collaborative filtering methods (accounting for slightly less than 40% and about 35%, respectively), but also reasoning or social networks are exploited in some cases.

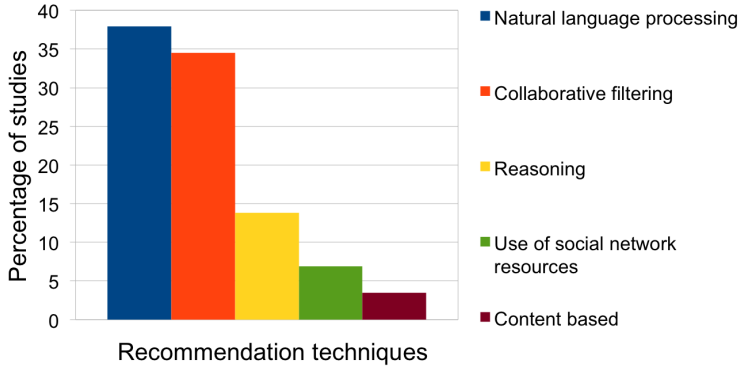


Fig. 3.4 Distribution of the hybrid studies according to the recommendation techniques that they exploit. Percentages refer to the total number of hybrid studies.

Dataset	**Number of studies**				
	General	LD driven	Hybrid	Hybrid and LD driven	LD driven only
DBpedia [12]	31	28	20	16	12
Freebase[6]	6	6	5	5	1
YAGO [50]	4	3	3	2	1
Wordnet [51]	4	2	3	2	0
DBLP [52]	3	3	3	3	0
Dataset independent	3	3	3	3	0
LinkedMDB [53]	3	3	3	3	0
Geonames[7]	2	1	2	1	0
MusicBrainz [54]	2	1	2	1	0
mySpace [55]	2	2	2	2	0
ACM[8]	1	1	1	1	0
IEEE[9]	1	1	1	1	0
Eventseer2RDF [56]	1	1	1	1	0
LinkedUp [57]	1	1	0	0	1
mEducator [58]	1	1	0	0	1
LinkedGeoData [59]	1	0	1	0	0
LODE [60]	1	1	1	1	0

Table 3.8 Distribution of the studies according to the Linked Data (LD) datasets that used.

[6] http://freebase.com/
[7] http://www.geonames.org/
[8] http://acm.rkbexplorer.com/
[9] http://ieee.rkbexplorer.com/

In addition, we studied which datasets are used and the outcome is presented in Table 3.8. It shows how many studies use a dataset overall and also by considering the four categories previously defined. It is possible to notice that DBpedia is used much more than the others. In fact, it is the biggest dataset and it is the most curated. Furthermore, it contains information about many different domains.

Other commonly used datasets are Freebase, YAGO, and Wordnet, but the latter is used in just half of the cases by Linked Data driven studies. In fact, it is also used with natural language processing techniques. On the contrary, the other datasets are used in most cases by Linked Data driven studies and often by studies which are both Linked Data driven and hybrid.

3.3.5 Application Domains

Figure 3.5 illustrates the application domains considered by the studies selected for the systematic literature review. The great majority of the studies (slightly less than 80%) focused on a single domain, while few (about 23%) can be used to recommend different kinds of items. An often occurring domain is music, which represents 17% and is followed by tourism and movies, accounting for roughly 10% each. Then web resources, expert recommendations, and video are between 5% and 7% each, and a number of other domains are considered by the remaining 10% of the studies.

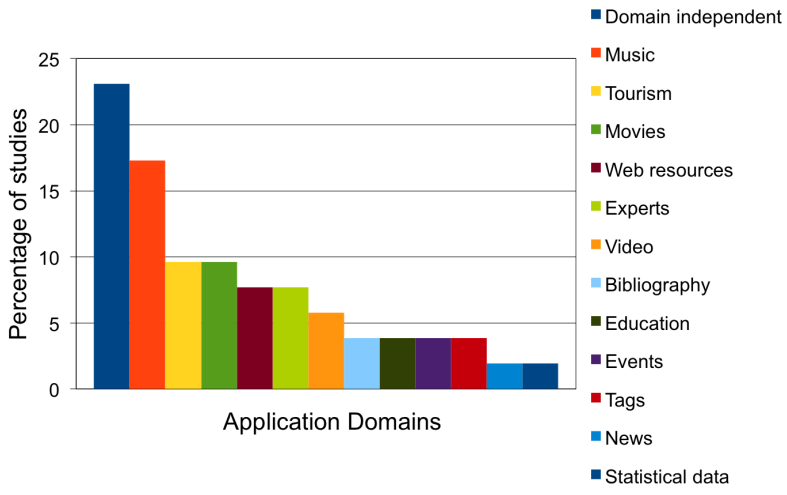


Fig. 3.5 Distribution of the studies selected according to the application domain.

3.3.6 Evaluation Techniques

RQ6 concerns RS evaluation, so we also dealt with this aspect. It is important to note that we focus on RS evaluation, thus GUI evaluation is not considered, although some of the studies addressed it. RS are commonly evaluated according to their computational complexity and accuracy [61]. The former measures the execution time required to produce recommendations, which depends on the complexity of the algorithms used as well as the runtime of third-party systems needed to produce recommendations. The latter is the capacity of the RS to satisfy the individual user's need for information and it can be evaluated by means of three techniques: online studies, user studies, and offline studies [22]. In online studies, recommendations are shown to users as they use the real-world system. Users do not rate recommendations, rather the system observes how often users accept a recommendation (e.g. through click-through rate). User studies involve users in order to compare recommendations generated by different algorithms with the users' judgments or ratings and the algorithm with the highest average rating is judged the best algorithm. Offline studies use pre-compiled datasets from which some information is removed for the evaluation. Subsequently, the algorithms are evaluated with respect to the capability to recommend the removed information. Usually, in any case, recommendations generated by a specific RS are compared with well-known similar approaches.[10] The most frequent measures are:

- Precision and recall, which evaluate the accuracy of RS taking into account the number of retrieved items, the number of items that evaluators considered as relevant and the total number of available items. The F-measure is the harmonic mean of precision and recall. Another precision measure is the Mean Average Precision (MAP), which is the average of the average precision of each user.
- User ratings, which are techniques in which lists of results from different RS are presented to users, who rate the lists according to their personal criteria [61].
- Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are metrics to measure the predictive accuracy of an RS in terms of rating prediction.

[10] A deep discussion of the methods and measures used to evaluate RS is out of the scope of this thesis. The reader may refer to the surveys of Beel et al. [61] or Shani and Gunawardana [22].

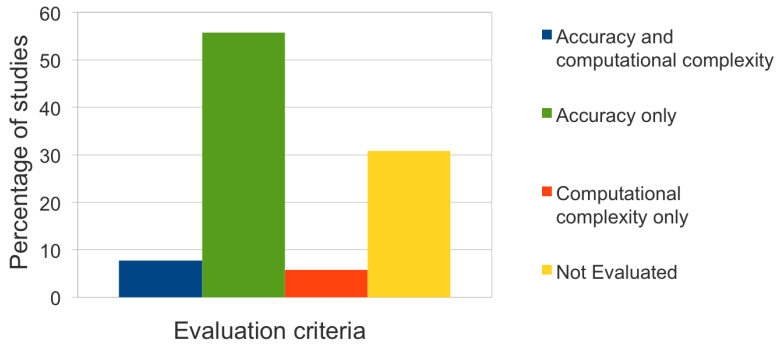


Fig. 3.6 Distribution of the studies selected according to the evaluation criteria which they use. Percentages refer to total number of studies.

MAE calculates the average absolute deviation between predicted similarities and similarity values in the real data set, while RMSE pays more attention to large errors [62]. Instead, ranking quality takes into account the retrieval correctness, which assigns an output ranking: a performance score based on the available reference relevance judgments [63]. Common metrics to measure the ranking quality are the Normalized Discounted Cumulated Gain (NDCG), average position and presence.

In addition, in an online study, the capability of explaining the provided recommendations was assessed, while in some user studies relevance and unexpectedness, i.e. the degree of novelty of a recommendation for the user, were also evaluated.

Figure 3.6 shows the main evaluation techniques found in the studies selected, as well as their classification and their occurrence in these studies. Studies which provided an evaluation accounted for about 70% of the studies included in the systematic literature review. Among these, roughly 55% only used an accuracy technique, while roughly 2% only evaluated the computational complexity and slightly less than 8% considered both accuracy and computational complexity.

Table 3.9 details the techniques used in the studies included by considering the type of evaluation used, the property evaluated and the measure used. The most frequent technique used to evaluate RS is measuring accuracy by means of precision and recall (used by 15 offline studies and 6 user study). We expected this

Type	Property	Measure	Number of studies
Online study	Explanation	User ratings	1
User study	Prediction accuracy	Precision	4
		Recall	2
		RMSE	2
		MAE	2
		MAP	1
		F-measure	1
		Others	6
	Relevance	User ratings	4
	Unexpectedness	User ratings	3
Offline study	Prediction accuracy	Precision	15
		Recall	15
		F-measure	4
		MAP	4
		NDCG	3
		RMSE	1
		Others	6
	Computational Complexity	Execution time	7

Table 3.9 Distribution of the studies selected according to the evaluation techniques used.

result because these metrics are the ones most commonly deployed in information retrieval approaches. Other widely used metrics are user ratings, applied in 8 studies, F-measure and MAP accounting for 5 studies each, together with execution time, which is exploited in 7 studies. We found only one work which relied on an online study, while user studies were used in 16 works in total and offline studies in 28. It is important to note that a study could use more than one type of evaluation (e.g. one user study and one offline study).

3.3.7 Future Work

RQ7 is related to directions for future research. To address this, we summarized the future work that the studies selected proposed in order to extend or improve their approaches. Specifically, about 67% of studies included in the systematic literature review present diverse proposals for future work. Table 3.10 lists the most important, indicating for each one, the number of studies in which it was mentioned. A deeper analysis of these results and a discussion of possible directions is presented in section 3.4.

Future work	Number of studies
Personalization of recommendations	8
Use more datasets	8
Create hybrid RS	7
Similarity measures	4
Find more semantic relationships (item-user and item-item)	3
Other proposals for future work	3
Consider other domains	2

Table 3.10 Distribution of the selected studies according to the future work that they propose.

In the following we provide a brief description of each item reported in Table 3.10:

Personalization of recommendations The idea is to know to what extent personalization can improve recommendations without requiring user profile information or user intervention for manual operations (feedback, filtering, annotation, etc).

Use more datasets It means to increase the range of data to annotate or match items to be recommended. It can also be useful to explore new domains because of the use of other datasets which can be from diverse domains.

Create hybrid RS This refers to exploring new ways to combine diverse recommendation techniques for creating hybrid approaches and improving the relevance and quality of recommendations.

Similarity Measures It is the creation of new similarity measures or the improvement of existing ones.

Find more semantic relationships It is the possibility of finding more semantic relationships between items and between users and items. It is considered by three studies.

Consider other domains Although domain dependency is one of the problems found in various studies, only two studies took into account exploring new application domains for providing recommendations.

Other proposals for future work This group includes applications in real life contexts, algorithms for categorization of recommendations, improving the performance of algorithms and the study of disambiguation techniques.

3.3.8 Limitations

The limitations reported in the selected studies are also related to RQ7 as these can help us to uncover the open issues in RS based on Linked Data and their relationships with proposals of future work. They are grouped into four main types: datasets, manual operations, personalization and computational complexity. We detail each of them in the following:

Datasets This type describes limitations of RS due to the datasets used. A number of studies required a copy of the entire dataset on a local server in order reduce the runtime to produce recommendations. This had to be done as sometimes public datasets offer limited results, restricted access and high timeout. Sometimes data had to be manually curated due to the poor reliability of public datasets. A number of RS are limited to the use of only one dataset. This can restrict the knowledge to which the RS can have access, avoiding data from diverse sources and domains being obtained.

Manual Operations It means that RS need the user to perform manual operations in order to produce recommendations. Some RS require manual selection of relevant concepts according to a specific application domain or interests. This is a difficult and tedious task considering the large amount of data that a typical Linked Data dataset can contain. Other RS did not rank their results, so final users are faced with no priority in the recommendation.

Personalization It is about producing recommendations according to the user profile or some personal features.

Computational complexity RS still need to improve the performance due to high computational demand to analyze large amounts of items and information stored into datasets. Another problem is the poor performance of public endpoints to access them.

3.4 Discussion

In the first part of this section we present a discussion of the results considering each research question, while in the second part we mention the limitations of our systematic literature review.

3.4.1 Specific Research Questions

This subsection discusses the research questions addressed in this systematic literature review according to the results reported in Section 3.3.

RQ1 is a general question regarding the studies that describe RS based on Linked Data. To provide an answer we have followed the steps described in the protocol presented in Section 3.2 in order to search and select studies in this area. Firstly, we retrieved a total number of 7873 papers (including those duplicated) from scientific digital libraries. After each researcher filtered papers by title and abstract, we discussed disagreements and we reach consensus on a final set of 69 papers to include in our study, which correspond to 52 unique studies.

RQ2 deals with research problems in the RS domain that researchers intended to solve by proposing approaches based on Linked Data. We found that the lack of semantic information and its complexity were the most notorious problems in RS.

Lack of semantics regards the need for rich semantic information about items. This is the main reason to devise novel strategies to represent items and user profiles using diverse semantic techniques exploiting several knowledge sources from the Linked Data cloud.

The complexity and heterogeneity of information and the subsequent cost of maintenance of knowledge bases makes Linked Data a suitable solution that uses publicly available knowledge bases that are continuously growing and maintained by third parties. However, this poses new challenges, for example, the need for mechanisms to assure the reliability of these knowledge bases that are used to describe user profiles and items and to generate recommendations.

Domain dependency is another problem that has been also addressed by using Linked Data because it allows the possibility to exploit information from different datasets that can be domain-independent or belong to diverse domains. In fact, this is one reason why the most used dataset is DBpedia as it is the most generic dataset that can be used for cross-domain RS. Nonetheless, some studies still report this problem as future work.

Computational complexity is a question that has not been widely addressed in the studies considered in this systematic literature review and remains an

open issue because most of the studies have concentrated only on semantic enrichment of items and inclusion of Linked Data datasets. Computational complexity needs to be addressed more because in RS not only accuracy is important, but also scalability and responsiveness. For example, this problem can be critical in RS for mobile scenarios where users demand fast response times.

Other problems such as usability, cold-start, data quality and data sparsity have been addressed by combining with Linked Data various techniques based on natural language processing, reasoning or social network resources and creating hybrid RS that exploit both collaborative filtering and content-based approaches.

RQ3 inquires about the contributions proposed in RS based on Linked Data. The analysis showed that the majority of studies are focused on providing new algorithms, but also on defining or extending a similarity measure of an ontology. Furthermore, adaptation, combination or extension to algorithms is quite often addressed together with information aggregation or enrichment. Accordingly, we found that Linked Data can be used in RS for several purposes such as:

- Defining different similarity functions between items or users by exploiting the large data available in the Linked Data cloud and the vast relationships already established such as properties or context-based categories. In this way, it is possible to extract semantic information from textual descriptions or other textual properties about the items in order to find semantic similarities based on the information stored in interlinked vocabularies of Linked Data. This can be useful in RS based on collaborative filtering to improve the neighborhood formation in user-to-user or item-to-item.
- Generating serendipitous recommendations, for example to recommend items that are not part of the user's personal data cloud, i.e. suggest new, possibly unknown items, to the user; or to guide users in the process of the exploration of the search space giving the possibility for serendipitous discovery of unknown information (for exploratory search systems).
- Offering the explanation of the recommendations given to the users by following the linked-data paths among the recommended items. In this

way, users can understand the relationship between the recommended items and why these items were recommended.

- Domain-independency when creating RS as it is possible to access data from Linked Data datasets from different domains.
- Enrichment of information sources such as databases, repositories, registries, etc. with information obtained from Linked Data datasets which manage huge amounts of open data. It offers the possibility to enrich graphs representing users and/or items with new properties in order to improve graph-based recommendation algorithms. Additionally, it helps to mitigate the new-user, new-item and sparsity problems.
- Annotating items and users with information from multiple sources facilitate RS to suggest items from different sources without changing their inner recommendation algorithms. Using such a semantic-based knowledge representation, recommendation algorithms can be designed independently from the domain of discourse.
- Obtaining hierarchical representation of items because of the topic distribution that some Linked Data datasets offer. In this way, RS can base their recommendation on the exploration of items belonging to similar categories.

RQ4 regards the diverse ways in which Linked Data is used to provide recommendations. First of all, we classified the studies according to the way they exploited Linked Data. As reported in Section 3.3, four categories were identified: *Linked Data driven RS* relies mainly on Linked Data to perform their tasks, *hybrid RS* uses Linked Data but also other techniques, *representation only RS* does not provide Linked Data based recommendations but it uses Linked Data for representing data based on RDF, and finally *exploratory search systems* that are not RS but may help users to find concepts or topics and have some similar features to RS especially in the use of Linked Data.

Table 3.11 describes each category including the most important studies that adopted these strategies, as well as their advantages and disadvantages. The numbers of the studies correspond to the identifiers indicated in Appendix A.

Most of the studies belong to the first category, and many belong to both the first and the second category. These two categories are also the most

Approach	Techniques	Advantages	Disadvantages
Linked Data-driven	- *Graph based*: weight spreading activation (S17), semantic exploration in an RDF graph (S29, S10, S3, S9, S19), and projections (S23) - *Reasoning*: (S1, S51) - *Statistical*: Matrix item-user (S29, S35, S31, S13, S37, S10), Scaling methods (S29) and topic discovery (S2)	- Generating serendipitous recommendations - Offering explanations of the recommendations following the linked-data paths - Creating domain-independent RS - Exploiting hierarchical information about items to categorize recommendations	- High cost of exploiting semantic features due to inconsistency of LD datasets - No personalization - No contextual information - High computational complexity - Need for manual operation - Need for dataset customization to address the computational complexity
Hybrid	- *Collaborative Filtering and Linked Data*: (S2, S4, S12, S25, S27, S3, S28, S26, S30, S35) - *Information aggregation and Linked Data*: opinions (S16), ratings (S19), and social tags (S32) - *Statistical methods and Linked Data*: Random Indexing (S10), VSM (S47, S31, S35), LDA (S35), Implicit feedback (S25), SVM (S13), Structure-based statistical semantics (S37)	- Overcoming the data sparsity problem - Allowing collaborative filtering RS to address the cold start problem	- High computational complexity
Representation Only	- Item/user information representation using RDF-based ontologies (S36, S38, S20, S40, S14, S15, S42, S46)	- Improving scalability and reusability of ontologies - Easing data integration - Enabling complex queries	- Difficult to reuse the already available knowledge in the Linked Data Cloud
Explorative Search	- Set nodes and associated lists (S49, S39, S34) - Spreading activation to typed graphs and graph sampling technique (S11)	- Enabling self-explanation of the recommendations	- No automation of the recommendation because explorative search approaches require frequent interaction with the user

Table 3.11 Classification of Linked Data based recommendation approaches.

interesting as they include RS to better exploit the advantages provided by Linked Data in order to reach best results. We also studied techniques to provide recommendations relying on Linked Data and slightly less than half of Linked Data driven RS used a dataset, almost one third define a similarity measure for Linked Data, while others adapt natural language processing or content-based methods or use reasoning.

With reference to the techniques used together with Linked Data, we found that natural language processing and collaborative filtering are the most used (both account for about one third of hybrid RS) as they intended to provide personalized suggestions of items tailored to the preferences of individual users.

Other techniques are less common (less than 15%) and they are reasoning, use of social network resources and content-based methods. Reasoning has not been widely used as its quality is still insufficient and its coverage is not enough broad at the level of system components and knowledge elements [64]. Therefore one solution is to develop RS based on reasoning-oriented natural language processing enriched with multilingual sources and able to support knowledge sources generated largely by people as Linked Data datasets.

As for the datasets used in the studies selected, we found that DBpedia is the most used Linked Data dataset. This may be because DBpedia is a generic dataset and most of the studies are domain independent that need to be evaluated in diverse scenarios. DBpedia is one of the biggest datasets that is frequently updated as it obtains data from Wikipedia that continuously grows into one of the central knowledge sources [65]. It makes DBpedia multimodal and suitable for RS that need to be domain independent and for knowledge-based RS where complexity and cost of maintenance of the knowledge base are high. However, for RS of a single domain, it could be better to use specific datasets, although always implementing a linking interface with generic datasets in order to resolve ambiguities or to exploit unknown semantic relationships.

RQ5 concerns the application domains considered by RS based on Linked Data so far. We identified 12 domains, but we found that some of the RS are domain independent (slightly more than one fifth of the studies). This is because most of the recommendation algorithms proposed can be applied in diverse domains by only changing the dataset or taking only a portion of it in order to obtain the

data to generate the recommendations. However, most of the studies (roughly 80%) targeted a single domain.

However, we also note that items of music, tourism and movies are the most recommended. This may be due to the large amount of data and state-of-the-art datasets available, which allow the researchers to compare their results with several works developed in the community.

Accordingly, in a number of cases the domain impacted also on datasets because they require a reduction of information, i.e, only a subset of concepts is considered, which requires offline processing and more effort to maintain the dataset even if it improves the performance. For example, Passant developed a RS named *dbrec* [66], which required to manually extract a subset of the data of DBpedia related with bands and musical artists.

RQ6 regards the evaluation techniques used to study RS based on Linked Data. Two main properties were considered in the study included: accuracy and computational complexity. Accuracy evaluates recommendations according to their relevance for final users, while computational complexity measures the execution time required to produce them.

With regard to accuracy, our results demonstrate that researchers rely more on offline study than user or online studies. This result was expected because offline studies require lower time and effort. Anyway, the usefulness of recommendations may depend on final user preferences more than comparing with similar approaches where evaluation may be biased as researchers must trust the results obtained. Although results of offline evaluations would not contradict results from online evaluations, there is also doubt on how reliable user ratings are [67, 68]. Therefore future methodologies of evaluation should be user-centered in order to assure the quality of the results of RS.

Additionally, as expected most of the studies selected were more likely to evaluate their recommendations applying traditional methods of information retrieval such as precision and recall that are focused on percentages of true positives, false negatives, and false positives.

Interestingly, we found that few works evaluated the computational complexity of RS, which is a critical factor especially for applications that need responses with short timeouts. Therefore it is still an open issue considering that accessing to Linked Data datasets in most cases is time-consuming and

requires that researchers download dumps of the datasets to access them in local repositories.

RQ7 aimed to uncover the most promising directions for future research on RS based on Linked Data. To address this issue, we have reported not only future works but also limitations of the studies selected.

Section 3.3.7 summarized the future work reported in the studies selected. We found that the most frequently mentioned future works were the personalization of recommendations, the use of more datasets, and the creation of hybrid RS.

The lack of personalization of recommendations is still a common drawback in Linked Data based RS. It concerns the fact that different users obtain the same set of results with the same input parameters. To solve this drawback some RS need explicit feed back from users in order to differentiate the results based on information about the user's profile (e.g. browsing history, favorite music genre, etc).

However these approaches force the user to perform extra work like rating items or building an exhaustive user profiles. Consequently, there is a need of non-invasive personalization approaches supported by Linked Data in order to obtain implicit information from the neighborhood relationships user-to-user, item-to-item and user-to-item. These relationships can be inferred from the links between concepts of Linked Data datasets related with properties of items and users. Using more datasets is needed in order to increase the base of knowledge to produce recommendations. As presented in Section 3.3.8, there are some limitations of the current Linked Data based RS with regard to the use of Linked Data datasets such as: restricted access, poor reliability, computational complexity, low coverage of languages, domain dependency and the need for installing a local copy of the dataset. For this reason, it is important to investigate new ways to integrate different datasets in order to: (*i*) extend the knowledge base allowing the RS to access to other datasets in case that the main dataset fails or the data are not reliable; (*ii*) create scalable RS because they can be adapted to other domains by only accessing to the appropriate dataset; (*iii*) and improve the performance by selecting datasets with better response time.

The creation of hybrid RS is not a new proposal, as could be seen in Section 3.3.4, combining diverse techniques of recommendation with Linked

Data-based approaches is a frequent practice in the studies selected. However, we also found that it is still an open issue because it is necessary to investigate which combinations of techniques are more suitable for a RS applied in diverse contexts. For example, combining Linked Data based RS with social-based RS can be a good choice for applications that require information about the users and their inter-relationships. In this way, RS can access information that sometimes is not available in Linked Data datasets such as items rating information, user profiles, and other social information.

The inclusion of user profile information (user profiling) is another aspect that is not widely considered in Linked Data recommender systems. The idea behind the user profiling is to obtain a meaningful concept driven representation of user preferences in order to enable more precise specifications of user's preferences with less ambiguity. Therefore, this can be also useful to contribute to the personalization of Linked Data based RS.

The automatic selection of the appropriate dataset according to the type of items or the application domain is another challenge that intend to improve the quality of recommendations. This dynamic process of selection can help the algorithms to choose the best strategy to find candidate items to be recommender based on the implicit knowledge contained in Linked Data and the relationships with properties of items and users.

As a consequence, it is also important to study new similarity measures and techniques able to automatically combine information from different datasets and to deal with the diversity of data in these datasets. Furthermore, it can be possible to create a statistical models of user interests to overcome the topical diversity of rated items.

Finally, we found that there is still a need for building testbeds in order to allow for rigorous, transparent, and replicable testing and for studying new techniques (or adaptation of those existing) for evaluating the accuracy and computational complexity of RS based on Linked Data. This also must consider that Linked Data based RS may access to large amounts of information and that links among items can be unknown to the users. Additionally, large-scale RS should be also evaluated in terms of the ability to scale and provide recommendations with data coming from millions of users/items.

3.4.2 Limitations of Our Systematic Literature Review

This section describes the main limitations we faced during our systematic literature review. Firstly, although some of selected papers were initially included because their title or abstract, in the end they were excluded because we could not access them from our university.

Secondarily, we only considered the most relevant paper for each study in order to calculate the frequency of problems, future work, contributions and evaluation techniques. As a consequence, we could be biased, as some papers belonging to the same study may present a problem or contribution not reported in the most relevant paper.

Finally, we did not perform deep validation. Due to time issues the majority of studies were read by one researcher, and cross-checking was performed only on about one third of the studies. Nonetheless, for some papers for which assessment was difficult there was discussion between the author and the other colleagues involved in the survey.

3.5 Conclusions

This systematic review has discussed 69 papers reporting 52 primary studies addressing Recommender Systems (RS) that make use of the structured data published as Linked Data. We identified the most relevant problems that these studies aimed to solve and summarized how they used Linked Data to provide recommendations. Although some of our results are already known, we have conducted a systematic study which provides evidences to support those results and limits possible bias and is to a degree repeatable because it follows a clearly defined protocol. Furthermore, we analyzed the contributions, limitations, application domains, and evaluation techniques of the selected studies to assess the reliability of their results, and the proposed directions for future research.

With regard to the research problems, we found that the most relevant ones were the lack of semantic information and the complexity of information about items. In order to overcome the lack of semantics, RS are enriched with diverse Linked Data datasets that are useful to describe users and items while reducing the ambiguity and exploiting the vast amount of links between related concepts stored in these datasets.

The majority of the studies selected have addressed these problems using Linked Data for several purposes, such as (*i*) finding new relationships or similarities based on links, paths, graphs and created on the basis of Linked Data; (*ii*) generating serendipitous recommendations, i.e. recommending items that are not expected by the users due the links uncovered once the items are enriched with Linked Data; (*iii*) and explaining the recommendations, i.e. allowing users to understand the reason of a recommendation by following the paths among items in the Linked Data cloud.

We also provided a classification of the studies selected according to the way they use Linked Data to provide recommendations. In particular we identified four classes: Linked data-driven RS, which rely on techniques applied on Linked Data datasets such as categories, paths, and number of input and output links; hybrid RS that combine traditional techniques of recommendation (e.g. collaborative filtering, content based, etc.) with Linked Data; representation only RS that uses Linked Data only to represent items or users but not for recommendations and exploratory search systems that are not RS, but help users to discover content through a guided search and are specially useful for users interested in learning or investigating a topic.

Additionally, we studied the most common datasets that RS use in order to obtain information and we found that more than a half of these studies rely on DBpedia. This may be because DBpedia is considered a central hub for the Linked Data cloud, i.e. it is linked to various datasets which gives the possibility to access to diverse data from different application domains. Additionally, it makes DBpedia suitable for testing purposes in generic RS.

Concerning the evaluation techniques the majority of the studies selected are focused on accuracy and rely more often on offline studies than user or online studies. Computational complexity is also assessed in few cases, however we think that it is an important factor to be evaluated especially for applications needing short responses such as RS in mobile environments. Additionally, we found there is still a need for building testbeds to allow for testing and studying the results of RS based on Linked Data.

According to our findings, we identified that two recurrent issues in the studies selected are the high computational demand and the domain dependency. Therefore, we believe that further research is still needed to offer non-invasive personalization, exploit more datasets and improve performance. Additionally, future work should

focus on providing evaluation of RS considering the accuracy and computational complexity. With regard to application domains music, movies and tourism items are the most used in RS and this may be due to the fact that in these domains there are more datasets which help scientists to assess the results of their RS in comparison with similar approaches.

Chapter 4

A Framework for Linked Data based Recommendation Algorithms

4.1 Introduction

Due to the increase of structured data published on the Web through the principles of Linked Data, one is more likely to find resources that describe or represent real life concepts. The information provided by these resources can be used in different domains. In particular, Linked Data may improve RS because they represent multi-domain knowledge, provide standard access to data, and represent semantic relationships among different entities [10]. However, finding and recommending related resources is still an open research area [1]. The problem of finding existing relationships between resources can be addressed by analyzing the categories they belong to, their explicit references to other resources and/or by combining both of these approaches. Currently, there are many works aimed at resolving this problem by focusing on specific application domains and datasets.

In this context, this chapter aims to answer the following research questions:

- *How can the existing algorithms for recommending resources from the Web of Data be compared to choose the one which best suits the characteristics of a given application domain and a given dataset?*
- *How can the performance and accuracy of the different existing algorithms be measured to select the one that best suits specific recommendation needs?*

To answer these research questions, we propose a framework for deploying and executing Linked Data based recommendation algorithms (implemented following some guidelines), which facilitates the conduction of studies for their evaluation in different application domains and without being bounded to a single dataset. Thus, the framework makes it possible to benchmark the algorithms in order to choose the one that best fits the recommendation requirements.

Additionally, the framework provides a set of APIs that enable application developers to use it as the main component for recommendations in a given architecture. In this way, developers do not need to deal with the execution platform of the algorithms but only focus their efforts either on selecting the existing algorithm that best fits their needs or on writing a customized one.

The remainder of this chapter is structured as follows: Section 4.2 introduces an evaluation framework for deploying recommendation algorithms. Section 4.3 details the framework including the main modules for discovering, ranking and categorizing resources. Finally, Section 4.4 presents the conclusions and future work.

4.2 The Allied Framework

Allied[1] is a framework to deploy and execute resource recommendation algorithms based on Linked Data. Through an implementation of these algorithms, it is possible to test them in different application domains and to analyze their behaviors.

Accordingly, the framework facilitates the comparison of the results for these algorithms both in performance and relevance. In this way, the framework creates an environment to select, evaluate, and develop algorithms to recommend resources belonging to different contexts and application domains that can be executed within the same environment and with different configuration parameters. Besides, it enables the creation of innovative applications on top of it. For studying recommendation algorithms, the recommendation process has been divided into four steps (as shown in Figure 4.1):

Resource generation. The first step is intended to generate a set of candidate resources (CR) that maintain semantic relationships with an initial resource (ir).

[1] http://natasha.polito.it/AlliedWI

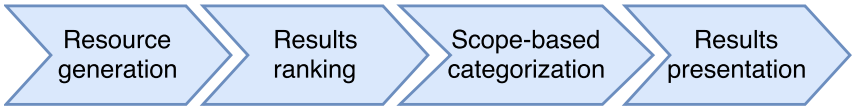


Fig. 4.1 Steps of the recommendation process.

The initial resource may be any resource identified with a URI. The semantic relationships may be seen as direct or indirect links between two resources in a Linked Data dataset.

Results ranking. It sorts the candidate resources generated in the previous step by considering the semantic similarity with the initial resource. In this step, different semantic similarity measures can be used to calculate the semantic similarity between pairs of resources.

Scope-based categorization. The list of ranked candidate resources generated in the previous step may be too general, that is, a recommendation may include resources from unrelated domains of knowledge. For this reason, this optional third step groups these resources already ranked into meaningful clusters that represent common knowledge domains.

Results presentation. Finally, the results of the last step are graphically presented through different facets to allow the end-users to visualize the recommendations.

Based on the recommendation steps mentioned previously, the architectural layers of Allied are: generation, ranking, classification, and presentation (as shown in Figure 4.2).

The recommendation process involves one or more layers of the framework. For example, Figure 4.2 shows the generation layer composed of a set of algorithms, $\{Rec_1, Rec_2, \ldots, Rec_n\}$ that retrieve resources located at a predefined semantic distance from an initial resource. These algorithms can be integrated with other algorithms of the same layer or of other layers.

Likewise, the algorithms of the ranking layer, $\{Rank_1, Rank_2, \ldots, Rank_n\}$ can be integrated with the generation layer algorithms in order to produce ranked lists based on the semantic relatedness between each tuple (ir, cr_i); where ir is the initial resource and cr_i is each one of the candidate resources generated by one of the $\{Rec_1,$

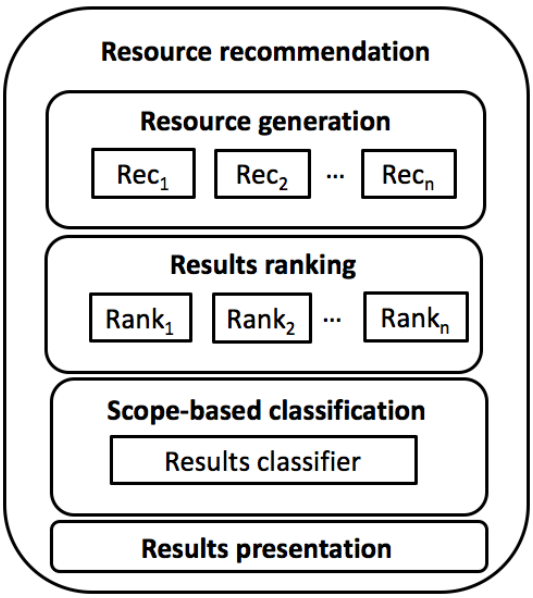


Fig. 4.2 The conceptual architecture of the Allied framework.

$Rec_2, \ldots, Rec_n\}$ algorithms. In this way, it is possible to produce recommendations based on semantic relationships (transversal, hierarchical or hybrid) and to study the application of these algorithms in different contexts and domains.

4.3 Implementation

The algorithms implemented for each layer of Allied are shown in Figure 4.3. Generation, Ranking, and Classification layers are responsible for the recommendation process, while Knowledge base core and Presentation are in charge of accessing the datasets and presenting the results.

4.3.1 Knowledge Base Core

This module represents the data layer of the Allied framework. It is the main data source containing knowledge about resources and their structural relationships. The current implementation of Allied uses the DBpedia dataset[2] as knowledge base, but it can be easily extended to other datasets. DBpedia was selected because it is a general

[2] http://dbpedia.org/

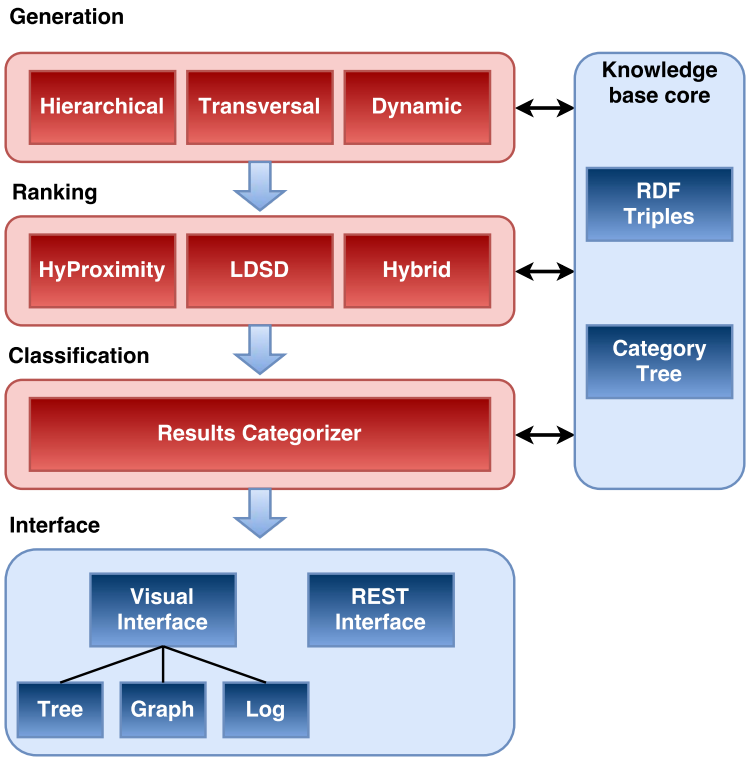


Fig. 4.3 The layered architecture of our implementation of Allied.

dataset that offers the possibility to evaluate the results in a number of scenarios. DBpedia is one of the biggest datasets that is frequently updated because its data comes from Wikipedia, and that continuously grows into one of the most interlinked datasets in the Web of Data [69]. Furthermore, a sub-module to retrieve the data from the datasets was developed using the RDF API Jena[3] for Java.

The knowledge base may be seen as a tuple *(R, T, L)* composed by resources (*R*), categories (*T*), and relationships (*L*), where:

Resources are instances of concepts identified by a URI. Concepts are abstractions from the real life like ideas or notions [70].

Categories denote types, concepts or classes and are the bases of the class hierarchy for the knowledge items. DBpedia provides information about the hierarchical relationships in three different classification schemata: Wikipedia categories,

[3] http://jena.apache.org/tutorials/rdf_api.html

YAGO categories[4] [50], and WordNet synset links[5] [51]. In this implementation, the Wikipedia categories (that are represented with concepts of the Simple Knowledge Organization System (SKOS) vocabulary [70]) were chosen to describe categories and their relationships. In effect, they are the most linked in DBpedia, consisting approximately of 80.9 million links for the year 2014.[6]

Relationships are the links (also known as properties) connecting resources or categories along the whole dataset graph. The knowledge base for the framework contains three types of relationships.

Resource-Resource (*R-R*) These are the transversal relationships between resources, which are those links between resources that do not refer to hierarchical classifications. Most of the links of DBpedia belong to this type.

Resource-Category (*R-T*) These are relationships between a resource and a category. They can be represented by using the RDF [11] property `rdf:type` or the SKOS properties `skos:subject` (hasCategory) and `skos:isSubjectOf` (IsCategoryOf). However, the two SKOS properties are deprecated [70] and consequently not used in DBpedia. Therefore, DBpedia relates resources to their Wikipedia categories using the `dcterms:subject` the Dublin Core vocabulary[7] instead. Accordingly, `dcterms:subject` is used in Allied for both relationships.

Category-Category (*T-T*) These are hierarchical relationships between categories within a hyponymy structure (a category tree). They can be represented by using the RDFS [40] property `rdfs:subClassOf` or the SKOS properties `skos:narrower` (isSuperCategoryOf) and `skos:broader` (isSubCategoryOf).

4.3.2 Generation Layer

This layer aims at discovering resources related to a given one through semantic relationships. Given an initial resource (or a set of initial ones) it generates a

[4] http://www.mpi-inf.mpg.de/yago-naga/yago/
[5] https://wordnet.princeton.edu
[6] http://wiki.dbpedia.org/Datasets#h434-7
[7] http://dublincore.org/documents/dcmi-terms/

set of candidate resources located at a predefined distance. For this layer, three generators were implemented based on the semantic relationships found on the Linked Data: (i) a transversal generator to study direct and indirect relationships between resources (Resource-Resource) avoiding hierarchical relationships, (ii) a hierarchical generator for indirect relationships between resources through direct relationships between resources and categories (Resource-Category) and between categories (Category-Category), and (iii) a dynamic generator which combines dynamically both types of relationships giving priority to the existing interlinking between resources. These generators use SPARQL [15] queries to navigate the dataset.

Transversal Generator

The transversal generator looks for resources that are directly related to a given initial resource and those found through a third resource (indirect relationships). Its implementation is inspired by dbrec [66].

```
SELECT DISTINCT ?cr WHERE {
    { <inURI> ?p ?cr . }
    UNION
    { ?cr ?p <inURI>. }
    FILTER(isURI(?cr)
        && ?p != <forbiddenLinkURI1>
        && ?p != <forbiddenLinkURI2>
        && ...
        && ?p != <forbiddenLinkURIn>).
}
```

Listing 4.1 The SPARQL query to retrieve resources directly linked to the resource <inURI>.

The SPARQL query used to retrieve the resources directly connected with the initial resource is presented in Listing 4.1. In this query <inURI> is the URI of the initial resource, p is the link and cr is each one of the candidate resources to be retrieved. A set of forbidden links can be defined to prevent the algorithm to obtain resources over links pointing to empty nodes (i.e. resources without a URI), literals that are used to identify values such as numbers and dates or nodes that are not desired for the recommendation. In other words, it is a way to limit the

results of the algorithm. For example the resource dbr:Turin contains the link <dbpprop:populationTotal> that points to the integer value 911823. Optionally, a set of allowed links may be added to restrict the set of retrieved resources to those linked with only a set of specific links. In the query of Listing 4.1, the forbidden links are limited adding the expression && ?p != <forbiddenLinkURI> for each link.

```
SELECT DISTINCT ?cr WHERE {
    { <inURI> ?p ?o .
    ?o ?p ?cr.}
    UNION{<inURI> ?p ?o . ?cr ?p ?o .}
    UNION{ ?o ?p <inURI>. ?o ?p ?cr .}
    UNION{ ?o ?p <inURI>. ?cr ?p ?o .}
    FILTER(isURI(?cr) && isURI(?o)
        && ?p != <forbiddenLinkURI1>
        && ?p != <forbiddenLinkURI2>
        && ...
        && ?p != <forbiddenLinkURIn>).
}
```

Listing 4.2 The query to retrieve resources indirectly linked to the resource <inURI>.

The SPARQL query to retrieve resources indirectly connected to the resource <inURI> through a third resource (o) is shown in Listing 4.2.

Hierarchical Generator

The hierarchical generator generates a set of candidate resources located at a specified distance in a hierarchy of categories taken from a category tree described in a dataset. The implementation of this module is inspired by the work of Damljanovic et al. [71], which obtains candidate resources by navigating a category tree of the Wikipedia categories.

The hierarchical generator firstly extracts base categories of an initial resource (<inURI>) and then looks for broader categories until a maximum distance (which may be user-defined) is reached. This maximum distance is the hierarchical distance of a broader category from base categories. It is inversely proportional to the level of specificity of a category (i.e. a higher distance means that a category contains

a lower level of specificity). Listing 4.3 presents the SPARQL query used for the hierarchical generator to obtain base categories of an initial resource (<inURI>). As said before, dcterms:subject is used in Allied because skos:isSubjectOf and skos:subject are deprecated and not employed in DBpedia.

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT ?cat WHERE {
    <inURI> dcterms:subject ?cat.
}
```

Listing 4.3 The SPARQL query to retrieve base categories of the resource <inURI>.

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?broaderCat WHERE {
    {<catURI> skos:broader ?broaderCat.}
    UNION
    { ?broaderCat skos:narrower <catURI>.}
    ?broaderCat rdfs:label ?categoryName.
    FILTER(lang(?categoryName) = "en").
}
```

Listing 4.4 The SPARQL query to retrieve broader categories of the category <catURI>.

Listing 4.4 shows the SPARQL query used to recursively extract broader categories for each base category starting from a distance equal to 1 until a maximum distance is reached. In this query, <catURI> is the URI of the sub category and FILTER limits the search for only categories in English language. After extracting categories, this module extracts subcategories for all the broader categories at maximum distance (i.e. it descends one level into the category tree) to increase the possibility of finding more candidate resources. Finally, the algorithm obtains candidate resources for each category (including subcategories). Listing 4.5 presents the SPARQL query that extracts subcategories of each broader category obtained by recursive application of the query shown in Listing 4.4. Listing 4.6 obtains candidate resources for each category. In this SPARQL query, <catURI> denotes a URI of one of the categories retrieved in previous steps to obtain related candidate resources. As a result, the module creates a "category graph", including the initial resource, its category tree,

and the candidate resources retrieved for each category. For example, Figure 4.4 shows the category graph for the resource Mole Antonelliana.

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?subCat WHERE {
    {<catURI> skos:narrower ?subCat.}
    UNION
    { ?subCat skos:broader <catURI>.}
    ?subCat rdfs:label ?categoryName.
    FILTER(lang(?categoryName) = "en").
}
```

Listing 4.5 The SPARQL query to retrieve subcategories of the category <catURI>.

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT ?cr WHERE {
    ?cr dcterms:subject <catURI> .
}
```

Listing 4.6 The SPARQL query to obtain candidate resources of the category <catURI>.

Dynamic Generator

The dynamic generator is a "hybrid" generator, which takes advantage of both the transversal and the hierarchical approaches, giving priority to the existing interlinking between resources, that is, one of the four principles of Linked Data [37]. The innovative algorithm of this generator is explained in Section 5.3

4.3.3 Ranking Layer

This layer mainly ranks candidate resources obtained in the previous layer, based on semantic similarity functions. These candidate resources are sorted according to values of a semantic similarity function, which measures the similarity between the initial resource and each one of these candidate resources. The framework in its current implementation includes (but is not limited to) three ranking algorithms.

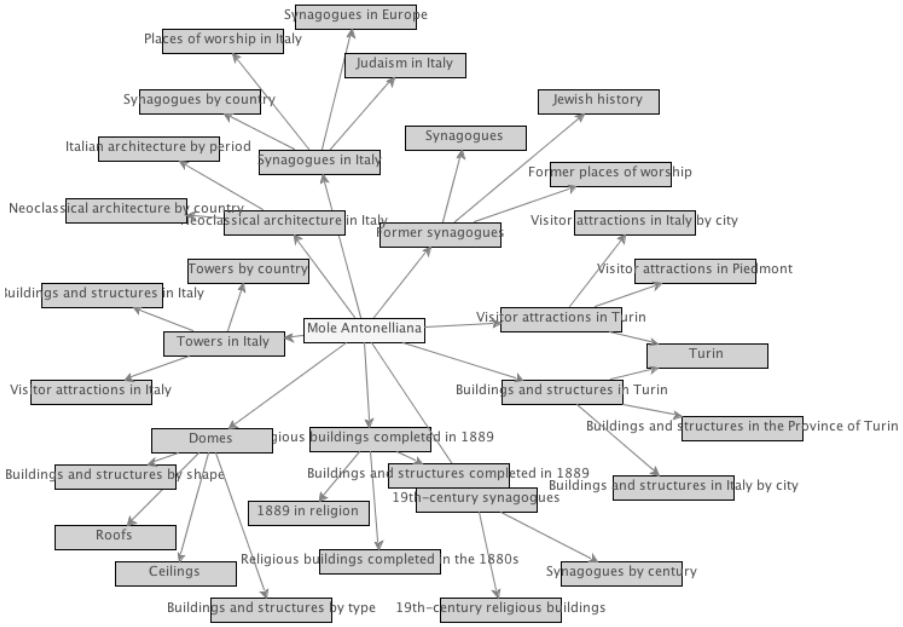


Fig. 4.4 Example of a category graph for the resource *Mole Antonelliana* (candidate resources are not included for space reasons).

Similarly, to the algorithms of the generation layer, the ranking algorithms are also based on the semantic relationships in Linked Data.

Transversal LDSD Ranking

The transversal LDSD ranking algorithm calculates the Linked Data Semantic Distance (LDSD) between an initial resource and each one of the candidate resources obtained in the generation layer. The LDSD distance, initially proposed by Passant [66], is based on the number of indirect and direct links between two resources. The similarity of two resources (r_1, r_2) is measured by combining four properties: the direct input links, the direct output links, the indirect input links, and indirect output links. Equation 4.1 illustrates the basic form of the LDSD distance.

$$LDSD(r_1, r_2) = \frac{1}{1 + Cd_{out} + Cd_{in} + Ci_{out} + Ci_{in}} \tag{4.1}$$

Cd_{out} is the number of direct input links (from r_1 to r_2), Cd_{in} is the number of direct output links, Ci_{in} is the number of indirect input links, and Ci_{out} is the number of indirect output links.

Unlike the implementation developed by Passant, which is limited to links from a specific domain, the LDSD function implemented in Allied takes into account all resources from the dataset. However, it can be customized to defined types of links belonging or not to a particular domain by adding a set of forbidden links.

The SPARQL query that counts direct input and output links between an initial resource <inURI> and a resource of the set of candidate resources is presented in Listing 4.7. The SPARQL query that counts input and output indirect links between an initial resource (<inURI>) and a resource (<crURI>) from the set of candidate resources is presented in Listing 4.8.

```
SELECT DISTINCT count(?p) WHERE {
    #output links
    { <inURI> ?p <crURI>. }
    #input links
    UNION
    { <crURI> ?p <inURI>. }
}
```

Listing 4.7 The SPARQL query to count input and output direct links.

```
SELECT DISTINCT count(?p) WHERE {
    #input links
    {?o ?p <inURI> . ?o ?p <crURI> .}
    UNION
    {?o ?p <inURI> . <crURI> ?p ?o .}
    #output links
    UNION
    {<inURI> ?p ?o . ?o ?p <crURI> .}
    UNION
    {<inURI> ?p ?o . <crURI> ?p ?o .}
}
```

Listing 4.8 The SPARQL query to count input and output indirect links.

Using these SPARQL queries, the transversal ranking algorithm calculates the LDSD for each pair of resources composed of an initial resource and each of the resources obtained from the generation layer.

HyProximity Ranking

The HyProximity ranking algorithm is based on the similarity measure defined by Stankovic et al. [72]. This measure can be used to calculate both transversal and hierarchical similarities. The HyProximity in its general form is shown in Equation 4.2 as the inverted distance between two resources, balanced with a weighting function. In this equation $d(r_1, r_2)$ is the distance function between the resources r_1 and r_2, while $p(r_1, r_2)$ is the weighting function, which is used to weight different distances.

$$hyP(r_1, r_2) = \frac{p(r_1, r_2)}{d(r_1, r_2)} \tag{4.2}$$

Based on the structural relationships (hierarchical and transversal), different distance and weighting functions may be used to calculate the HyProximity similarity:

Hierarchical Hyproximity The definition of this similarity function relies on the work of Stankovic et al. [72]. It depends on the maximum distance of categories from the initial resource as in the hierarchical generator algorithm (Section 4.3.2). In particular, $d(ir, r_i) = maxDistance$, where ir is the initial resource, r_i is each one of the candidate resources generated in the hierarchical algorithm, and $maxDistance$ is the maximum distance of the broader categories from the base ones. The weighting function is defined in Equation 4.3, which is an adaptation of the informational content function defined by Seco et al. [73]. In this equation, $hypo(C)$ is the number of descendants of a category C and $|C|$ is the total number of categories in the category graph of C. This function was chosen as minimizes the computation complexity of the informational content regarding to other functions that use external corpora [74].

$$p(C) = 1 - \frac{\log(hypo(C) + 1)}{log(|C|)} \tag{4.3}$$

Transversal Hyproximity In this similarity function $d(ir, r_i) = maxDistance$ if the generator of resources is hierarchical, otherwise $d(ir, r_i) = 1$ for resources connected to the initial resource through direct transversal links or $d(ir, r_i) = 2$ for indirect transversal links. The weighting function is defined in Equation 4.4: $p_{transv}(r_1, r_2)$ depends on the number of resources connected over a specific property (n) and the total number of resources of the dataset (M). Nonetheless,

in Allied, this algorithm is not limited to a specific property, and optionally can be configured to support a set of forbidden links or allowed links in a similar way as shown in Section 4.3.2 for the generation layer. The number of direct and indirect links was calculated with SPARQL queries. The value of M was fixed to the number of resources contained in DBpedia.[8]

$$p_{transv}(r_1, r_2) = -\log \frac{n}{M} \tag{4.4}$$

4.3.4 Classification Layer

Since this implementation of the framework is based on DBpedia, which is a general-purpose dataset, the results obtained may contain an inherent ambiguity due to the generality of the data used to produce recommendations. Moreover, a single ranked list of recommendations may not always be a good way to show this kind of general results because users may require results arranged according to their personal needs or knowledge domain. For this reason, the classification layer provides mechanisms to group the results obtained from the ranking layer into meaningful clusters that represent domains of knowledge.

Currently, the classification layer relies on Algorithm 1, which is the only algorithm implemented in Allied for categorizing resources based on the hierarchical relationships that exist on the Web of Data. As a result, when an application requires to classify resources according to a knowledge domain, the classification algorithm provides a mechanism to easily access the recommended items organized by clusters. Although, in the current implementation of Allied the resulting clusters correspond to Wikipedia categories, the approach could be extended to allow the definition of custom clusters by aggregating a number of categories or rely on other category schemas from the Web of Data, such as YAGO classes.

Algorithm 1 receives as input a set of ranked candidate resources (CR), an initial resource $inURI$, a maximum distance ($maxDistance$), and optionally an initial category graph, Gc_{in} (in case that a hierarchical structure is already available). If Gc_{in} is not given, then the algorithm creates a new category graph Gc containing categories for the initial resource and the set of candidate resources until a maximum distance ($maxDistance$). Otherwise the algorithm creates a copy of Gc_{in} (Lines 1 - 5).

[8] http://wiki.dbpedia.org/Datasets#h434-7

Algorithm 1 The hierarchical classification algorithm

Require: CR, $inURI$, $maxDistance$, optionally Gc_{in}
Ensure: A category graph Gc
1: **if** $Gc_{in} = null$ **then**
2: $Gc = createCategoryGraph(CR, maxDistance)$
3: **else**
4: $Gc = Gc_{in}$
5: **end if**
6: $C_{maxDistance} = getMaxDistanceCategories(Gc)$
7: **for each** pair of categories $(c_i, c_j) \in C_{maxDistance}$ **do**
8: $c_{lcb} = getLowestCommonBroaderCategory(c_i, c_j)$
9: Add c_{lcb} to Gc
10: Add $edge(c_i, c_{lcb})$ and $edge(c_j, c_{lcb})$ to Gc
11: **end for**
12: $intersectCategories(Gc)$
13: $deleteEmptyCategories(Gc)$
14: **return** Gc

In this implementation, *maxDistance* is set to 2 because some experiments showed that it is a reasonable trade-off between the number of categories and the time consumed. Afterwards, the algorithm extracts categories at the highest distance ($C_{maxDistance}$) and creates pairs of categories combining the elements of $C_{maxDistance}$ (Lines 6 - 7). Next, the function *getLowestCommonBroaderCategory* is executed to find a set of broader categories subsuming the categories of the set $C_{maxDistance}$. These new broader categories are then added to Gc including their edges ((c_i, c_{lcb}) and (c_j, c_{lcb})) (Lines 8 - 11). This function retrieves the shared ancestor that is more specific, i.e. closer to the two initial nodes or, equivalently, farthest from the root of the category tree. This is an instance of the lowest common ancestor problem in a tree or directed acyclic graph, also known as the least common subsumer in ontologies, which applies to the category tree in our case.

Finally, the updated set of categories of Gc are intersected and a function *deleteEmptyCategories* is executed to remove from the graph those categories subsuming less than three subcategories (i.e. only categories c_i, c_j). In this way a classification of higher distance for the candidate resources is created (Lines 12 - 14).

4.3.5 Presentation Layer

Allied can easily be integrated to any application that requires recommendations based on Linked Data. The current implementation includes three main interfaces that provide mechanisms to present results to the final user: a web interface, a standalone interface, and a RESTful interface. Figure 4.5 shows the main view of Allied web interface that allows the user to choose a recommendation algorithm. Note that this version is limited to predefined configurations of generation and ranking algorithms, representing state-of-the-art approaches, although more combination are possible in Allied. This has been done because we aimed to provide a prototype which do not require a complex set-up.

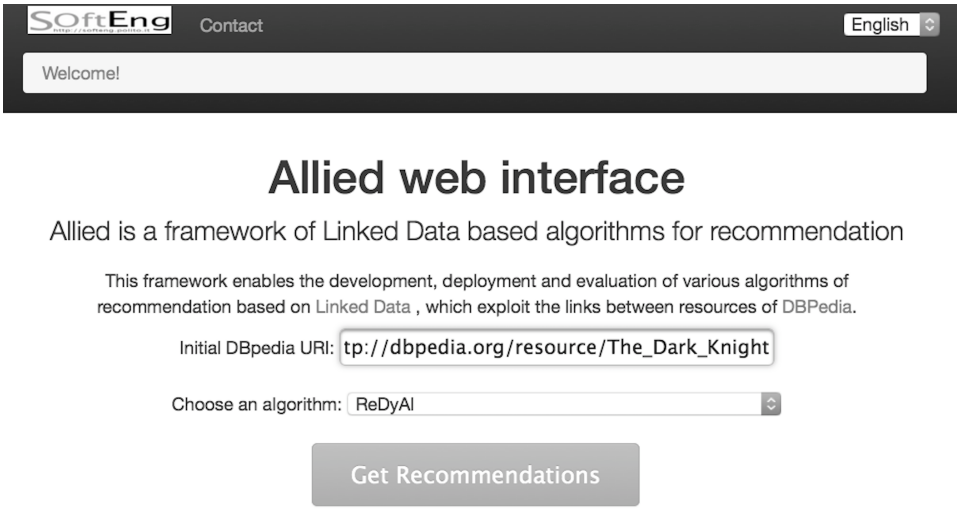


Fig. 4.5 The home view of the Allied web interface.

Figure 4.6 depicts an example of some results presented in the Allied web interface for the initial resource The Dark Knight, which is a movie. For space reasons, only the first five results are shown in the example. Figure 4.7 illustrates the setup view for the desktop version of the Framework. This view allows the user to choose the implemented algorithms for generation and ranking, and to select configuration parameters for the execution, e.g. the hierarchical distance, which is the maximum distance for the hierarchical generator and ranking. Additionally, this view allows the user to choose how he/she wants to see the results: as a graph or a tree. Figure 4.8 shows an example of a tree of results for the resource Mole Antonelliana: in this case, candidate resources are arranged in folders that represent clusters. An example of a graph of results is depicted in Figure 4.4.

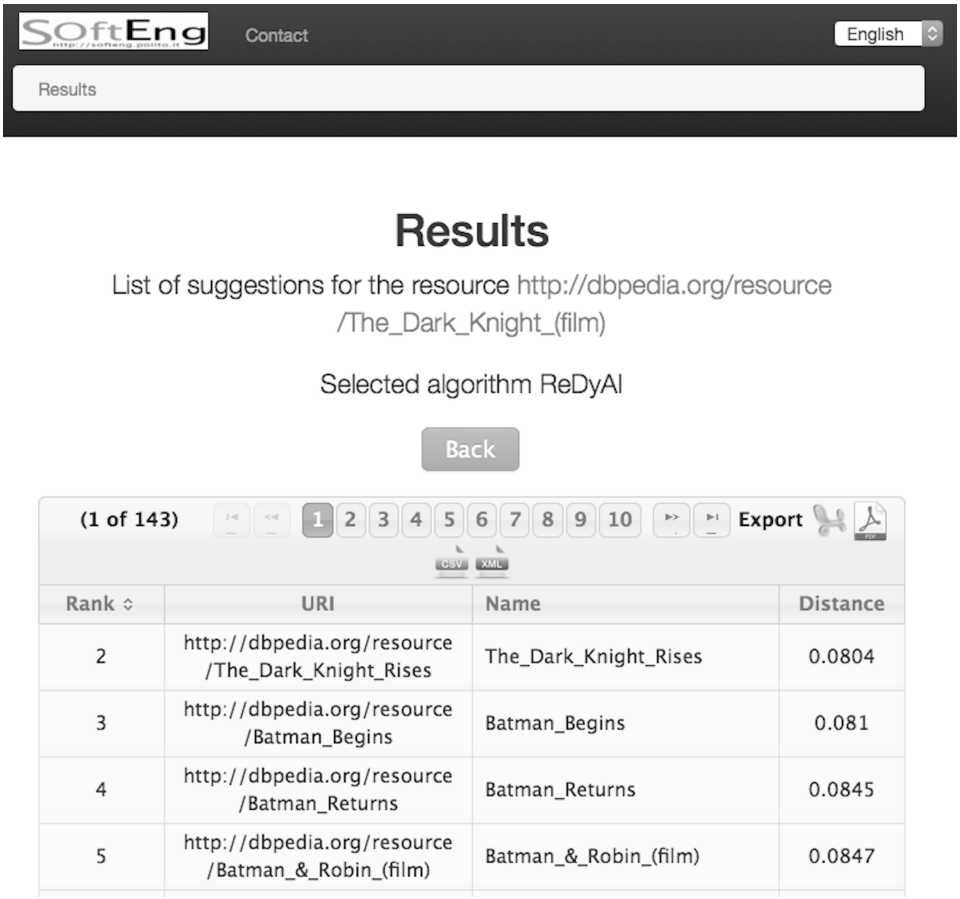


Fig. 4.6 An example of results shown in the Allied web interface.

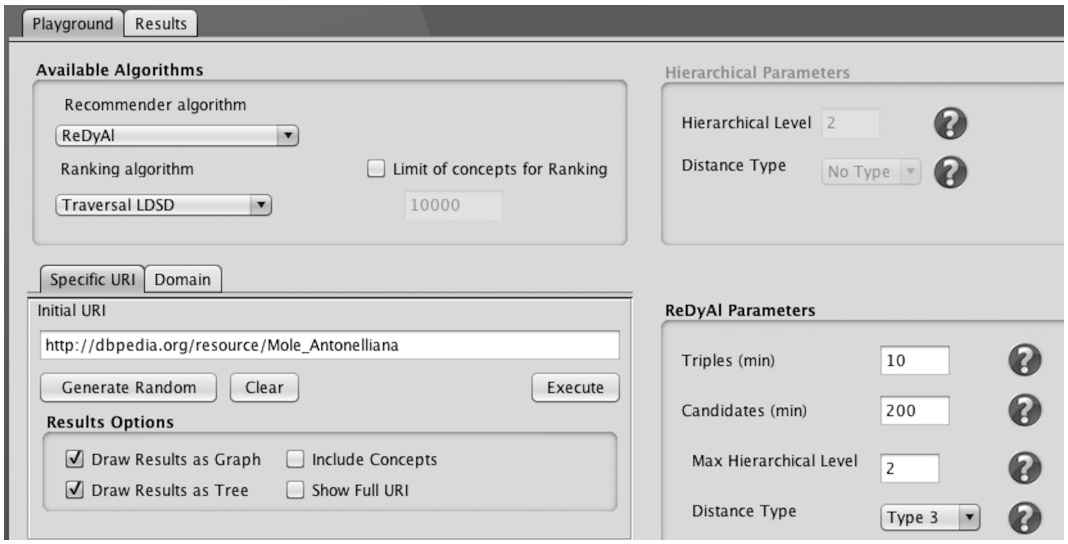


Fig. 4.7 The set-up view for the desktop version of the Allied framework.

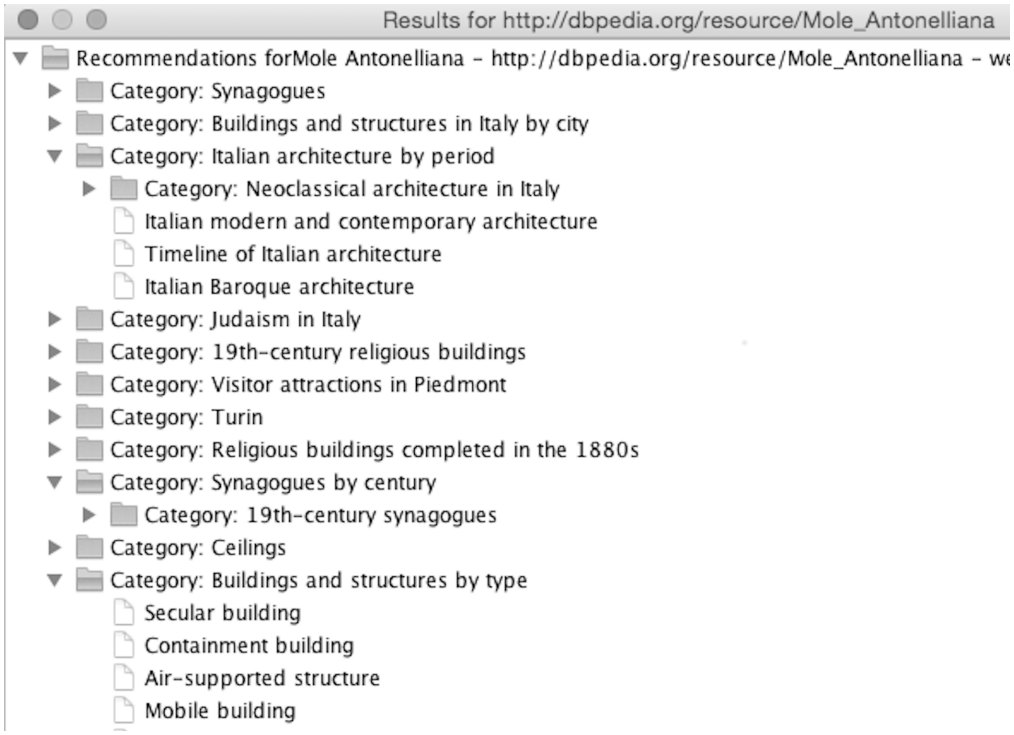


Fig. 4.8 An example of results as tree view for the desktop version of the Allied framework.

The RESTful interface is publicly available on the Web. It is based on two parameters: `uri` and `scope`. The former indicates the URI of the initial resource, while the latter is optional and enables to limit the recommendations to resources belonging to the specified cluster. If it is missing, a list of clusters of the initial resource is returned by the service. An example of an HTTP request to retrieve the clusters of Mole Antonelliana is presented in Listing 4.9, while the request to retrieve the resources related to Mole Antonelliana which are located in Turin is shown in Listing 4.10

```
GET /recommendations
    ?uri=http://dbpedia.org/resource/Mole_Antonelliana
```

Listing 4.9 An HTTP request to retrieve the clusters of Mole Antonelliana.

```
GET /recommendations
    ?uri=http://dbpedia.org/resource/Mole_Antonelliana
    &scope=http://dbpedia.org/resource/Category:Turin
```

Listing 4.10 An HTTP request to retrieve the resources related to Mole Antonelliana which are located in Turin.

4.4 Conclusions

This chapter presented Allied, a framework for deploying and executing recommendation algorithms that use Linked Data as their knowledge base.

The current version of Allied implements a set of three state-of-the-art transversal and hierarchical algorithms and ReDyAl, an hybrid algorithm that dynamically integrates both the transversal and hierarchical approaches for discovering resources that is presented in Chapter 5. The framework enables to choose the best algorithm implemented for recommending resources from the Web of Data when focusing on a specific application. It also enables to select the algorithm which best suits for a particular domain because it provides cross domain recommendations, since it relies on DBpedia. In addition, since the approach exploited is general, it is possible to adapt Allied to other datasets and select the algorithm which best fits the characteristic of the dataset.

The algorithms currently implemented with Allied were evaluated and compared by conducting a user study which is presented in Section 5.4 and relying on Allied. This framework facilitated the study, since the algorithms were deployed in the same environment and the generated recommendations were aligned. Thus, it enabled to measure and compare the accuracy of the algorithms. For this reason, conducting studies through our framework may also increase their reproducibility. At the moment it is possible to test the framework through its web interface.[9]

[9] http://natasha.polito.it/AlliedWI

Chapter 5

A Dynamic Recommendation Algorithm Based on Linked Data

5.1 Introduction

The work presented in this chapter holds on the results obtained from the study presented in Chapter 3 and is its continuation. The study stated that the problem of finding existing relationships between resources could be addressed by analyzing the categories they belong to, their explicit references to other resources and/or by combining both these approaches. The study also showed that many works aimed at resolving this problem by focusing on a specific application domain and dataset. In this chapter, we address this issue, and we focus on the following research questions:

- *How can we design a recommendation algorithm that exploits existing relationships between resources on the Linked Data, is independent of the application domain and may be used on different datasets on the Web of Data?*
- *How can we design a recommendation algorithm that provides novel recommendations, i.e., recommendations of resources not previously known to the user, without affecting prediction accuracy?*

We propose a new algorithm based on Linked Data which exploits existing relationships between resources to recommend related resources. It dynamically analyzes the categories they belong to and their explicit references to other resources,

then combines the results. The algorithm has been applied to DBpedia,[1] but it could as well be applied to other datasets on the Web of Data, and it is not bound to any particular application domain.

We conducted a user study to comparatively evaluate its accuracy and novelty against three state-of-the-art algorithms, which showed that our algorithm provides a higher number of novel recommendations while keeping a satisfying prediction accuracy. An implementation of our recommendation algorithm has been integrated into two mobile applications, which were developed in collaboration with Telecom Italia, the major network operator in Italy. The first suggests movies, while the second assists tourists. Both are based on DBpedia.

The chapter is organized as follows: Section 5.2 reviews related works; Section 5.3 presents our algorithm; Section 5.4 describes the evaluation method and provides the results; Section 5.5 shows the application of our algorithm for recommending movies and tourist attractions; Section 5.6 provides the conclusions.

5.2 Related Work

The different approaches to exploit Linked Data to recommend resources are described in Chapter 3. Some studies infer relationships between resources by taking into account the existing links between them in a dataset and use these relationships to measure the semantic similarity of the resources. Such relationships can be direct links, paths, or shared topics between sets of items. The most important related works are summarized in the following.

Damljanovic et al. [71] recommended experts in an open innovation scenario. Their approach, named *HyProximity*, takes as input a description of a problem in natural language and extracts a set of relevant words that are linked with resources of DBpedia. Then it generates recommendations by combining two techniques. The first one discovers resources related through hierarchical relationships, while the second one is based on transversal relationships, which connect resources without establishing a classification or hierarchy. By exploiting these two kinds of relationships, the approach identifies a set of direct or indirect topics related with potential experts to solve an innovation problem.

[1] http://dbpedia.org

Passant [66] described *dbrec*, a recommender system targeted for the music domain, which mainly relies on a distance measure named Linked Data Semantic Distance (LDSD). It takes into account the number of direct or indirect links between resources (related to the music domain) represented in DBpedia. Unlike HyProximity it does not distinguish between transversal and hierarchical links. Both Damljanovic et al. and Passant had to reduce the set of resources and links of the dataset to those belonging to a single domain (innovation problems and music respectively), which involves a massive effort to define which resources or links should be considered.

Other works combine Linked Data based algorithms with other techniques of recommendation to improve the results. These techniques include collaborative filtering [75–78], information aggregation [79–81] and statistical methods like Vector Space Model (VSM) [62, 77], Random Indexing (RI) [72], implicit feedback [77], Latent Dirichlet Allocation (LDA) [82], and structure-based statistical semantics [83]. De Graaff et al. [84] proposed a knowledge-based recommender system that derives the user interests from the user's social media profile, which is enriched with information from DBpedia. Musto et al. [85] compared several techniques to automatically feed a graph-based recommender system with features extracted from Linked Data. However, these methods usually require additional information from the user to produce accurate recommendations.

We propose a new recommendation algorithm, which is cross-domain and cross-dataset. It relies only on Linked Data and does not require to reduce the set of resources and links of the dataset to those belonging to a specific domain.

5.3 ReDyAl

ReDyAl is a recommendation algorithm which takes into account the different types of relationships between the data published according to the Linked Data principles. It aims at discovering related resources from datasets that may contain either *well linked* resources as well as *poorly linked* resources. A resource is said to be well linked if it has more links than the average number of links per node in the dataset; otherwise, it is poorly linked. The algorithm can dynamically adapt its behavior to find a set of candidate resources to be recommended, relying on the implicit knowledge contained in the Linked Data relationships.

5.3.1 Principles

As explained in Section 4.3.1, any dataset on the Web of Data may be seen as a tuple *(R, T, L)* composed of resources (*R*), categories (*T*), and relationships (*L*). Categories denote types, concepts or classes. Resources are instances of concepts identified by a URI. Relationships are also known as links or properties and connect resources or categories along the whole dataset graph. Categories often are hierarchically organized. For example, DBpedia provides information about hierarchical relationships in three different classification schemata: Wikipedia categories, YAGO classes[2] [50], and WordNet synsets[3] [51]. Relationships can be of three types: Resource-Resource (*R-R*), Resource-Category (*R-T*), and Category-Category (*T-T*). Considering this model of a dataset, ReDyAl consists of three stages:

1. The first stage discovers resources by analyzing the links between the given initial resource and other resources. Only R-R relationships are considered at this stage, although they can be indirect, i.e. they can connect two resources through a third one.

2. The second stage analyzes the categorization of the given initial resource and discovers similar resources located in the same categories. It finds indirect relationships between resources through direct R-T and T-T relationships. It is possible to specify to the algorithm which specific R-T and T-T relationships to consider in this step: the choice for R-T relationships is between `dcterms:subject` or `rdf:type`, while `skos:broader` and `skos:narrower` or `rdfs:subClassOf` are acceptable T-T relationships.

3. The last stage intersects the results of both the previous stages and ranks them by giving priority to those found in the first stage. The algorithm computes the similarity of the initial resource with any of the discovered resources, based on a similarity function which combines the Linked Data Semantic Distance (LDSD) [66] and HyProximity distance [71], opportunely adapted and generalized.

The algorithm can be applied to any dataset in the Web of Data. In the first step, it relies only on R-R relationships: any relationship of this kind may be used,

[2] http://www.mpi-inf.mpg.de/yago-naga/yago/
[3] https://wordnet.princeton.edu

independently of the data stored in the dataset. In the second step, the algorithm can be configured to use the `dcterms:subject` or `rdf:type` properties, which are R-T relationships. DBpedia uses both to enable different categorizations; for example to rely on the Wikipedia categories, it is necessary to set `dcterms:subject` as R-T relationship and `skos:broader` and `skos:narrower` as T-T relationships. Any other dataset uses at least `rdf:type` to indicate the class which a resource is an instance of. Thus, `rdf:type` can be used to find resources in the same class and then `rdfs:subClassOf` can be used to retrieve more general classes (or `skos:broader` and `skos:narrower`, if the categories are organized through SKOS properties).

The algorithm is independent of the application domain because it relies only on R-R, R-T or T-T links. If there are relationships among resources in different domains the algorithm may generate cross-domain recommendations. For example, DBpedia is a general dataset which represents resources of different kinds, and there may be a relationship between a song and a city because the song was recorded in that city or is about the city. Alternatively, there may be a link between a song and a movie because the song was part of the soundtrack of the movie. Thus, a city or a movie may be recommended starting from a song. R-T links may also generate cross-domain recommendations if resources which belong to different domains are included in the same category.

5.3.2 Reducing the Search Space

Additionally, the algorithm may be configured with a set of forbidden links to restrict the kind of links the algorithm should consider. This is useful to prevent the algorithm to obtain resources over links pointing to empty nodes (i.e. resources without a URI), literals that are used to identify values such as numbers and dates, and other nodes that are not desired for the recommendation. In other words, it is a way to limit the results of the algorithm. For example, the DBpedia resource `dbr:Turin` contains the link `dbpprop:populationTotal` that points to the integer value 911823: we can configure this link as forbidden link since it does not point to a resource which can be recommended. This is also useful to increase the performance of the algorithm because limiting the number of results decreases the ranking time. All the links which are not explicitly specified as forbidden are allowed links and define a domain of interest. This may be useful when the algorithm is applied to a generic dataset as DBpedia. This dataset contains millions of links between resources,

and if a developer is creating an application in the music domain then he/she may be interested only in resources of that domain, so he/she may want to consider only links pointing to those resources, i.e., a set of allowed links. In fact, the algorithm is cross-domain; thus it may recommend a city or a movie starting from a song, as we have already explained. While this may be an advantage in some applications, in others it may be confusing, especially if not adequately explained to the user. To limit the recommendations to specific categories of resources (for example to consider only tracks and artists), it is sufficient to “allow" only the relationships which point to these kinds of resources, i.e. have such desired category as the range.

5.3.3 Parameter Settings

ReDyAl receives as input an initial resource by specifying its corresponding URI (*inURI*), and three values (*minT*, *minC*, *maxDistance*) for configuring its execution. The selection of *minT* and *minC* is arbitrary and depends on the dataset and the convenience of the user who is setting up the algorithm. *minT* is the minimum number of links (input and output links involving the initial resource) necessary to consider a resource as well linked. The proper value of *minT* depends on the dataset: if it contains resources with a high number of links between them it is expected to be higher, while if the resources have only a few links, it should be set to a lower value. However, this parameter impacts on the algorithm: if the initial resource is well linked, transversal interlinking has a higher priority in the generation of candidate resources. Otherwise, the algorithm gives priority to the hierarchical relationships. For example, a user may consider the use of the hierarchical algorithms only if the resources are connected with less than ten links by setting *minT* to 10. In a similar way, the user may arbitrary fix the value of *minC*, which is the minimum number of candidate resources that the algorithm is expected to generate, i.e. the number of candidate resources the user is expecting.

The value of *maxDistance* limits the distance (i.e. the number of hierarchical levels) that the algorithm considers in a category tree. *maxDistance* may be defined manually; this is particularly useful when there are not enough candidate resources from the categories found at a certain distance (i.e. the number of candidate resources retrieved is lower than *minC*). In this case, the algorithm increases the distances to find more resources and if the *maxDistance* value is reached with less than *minC* candidate resources, the algorithm ranks only the candidate resources found until

that moment. Additionally, the algorithm may receive a list of forbidden links (FL) to avoid searching for candidate resources over a predefined list of undesired links.

5.3.4 Algorithm

ReDyAl (Algorithm 2) starts by retrieving a list of allowed links from the initial resource. Allowed links are those that are not specified as forbidden (FL) or that are explicitly defined in the initial resource. If there is a considerable number of allowed links (more than $minT$, i.e., the initial resource is well linked) the algorithm obtains a set of candidate resources located through direct (DRl_k) or indirect transversal links (IRl_k). This is done starting from the links explicitly defined in the initial resource (Lines 1-8). A resource is indirectly linked to the initial resource if it is linked through another resource. A resource directly linked is located at transversal distance 1 from the initial resource, while a resource indirectly linked is located at transversal distance 2 from the initial resource. With regards to the transversal links, a maximum distance of 2 is considered because for distances higher than 2 (i.e., one direct heap plus one indirect heap) the number of retrieved resources is dramatically increased, therefore increasing also the number of resources that are not relevant or related to the initial resource.

Next, if the current number of candidate resources generated (CR_{tr}) is greater than or equal to $minC$, the algorithm terminates returning the results (Lines 9-10). Otherwise, the algorithm generates a category graph (Gc) with categories of the first distance and applies iterative updates over the category graph over n distances from the initial resource, obtaining broader categories (i.e. more generic categories that are located in a higher level in a classification) until at least one of two following conditions is fulfilled: the number of candidate resources is sufficient ($|CR| > minC$), or the maximum distance is reached ($currentDistance > maxDistance$). At each iteration, candidate resources (CR_{hi}) are extracted from the broader categories of maximum distance (Lines 14-23). In any case, the algorithm combines these results with the results obtained in Lines 3-8 (adding CR_{tr} and CR_{hi} to CR). Finally, the set of candidate results is returned (Line 23).

Algorithm 2 ReDyAl algorithm

Require: $inURI$, $minT$, $minC$, FL, $maxDistance$,
Ensure: A set of candidate resources CR

1: $L_{in} = readAllowedLinks(inURI, FL)$
2: **if** $|L_{in}| \geq minT$ **then**
3: **for all** $l_k \in L_{in}$ **do**
4: $DRl_k = getDirectResources(l_k)$
5: $IRl_k = getIndirectResources(l_k)$
6: Add DRl_k to CR_{tr}
7: Add IRl_k to CR_{tr}
8: **end for**
9: **if** $|CR_{tr}| \geq minC$ **then**
10: **return** CR_{tr}
11: **else**
12: $currentDistance = 1$
13: $Gc = createCategoryGraph(inURI, currentDistance)$
14: **while** $currentDistance \leq maxDistance$ **do**
15: $CR_{hi} = getCandidateResources(Gc)$
16: **if** $|CR_{hi}| \geq minC$ **then**
17: Add CR_{tr} and CR_{hi} to CR
18: **return** CR_{hi}
19: **end if**
20: increase $currentDistance$
21: $updateCategoryGraph(currentDistance)$
22: **end while**
23: Add CR_{tr} and CR_{hi} to CR
24: **end if**
25: **end if**
26: **return** CR

5.3.5 Ranking of the Recommended Resources

The final operation is ranking the sets of candidate resources. The ranking process receives as input the candidate resources retrieved by the ReDyAl algorithm and ranks them according to their degree of similarity with the initial resource. This similarity is computed based on a combination of two distance measures: LDSD and HyProximity, which have been presented in Section 4.3.3.

The measure that combines LDSD and HyProximity used by ReDyAl is defined in Equation 5.1, where α and β may be set according to the convenience of the user: α is the weight for the transversal algorithm and β is the weight for the hierarchical

algorithm. In this way, resources are ranked in descending order, arranged from the largest to the smallest value of $Hybrid_{sim}$.

$$Hybrid_{sim} = (1 - LDSD)\alpha + (hyP(r_1, r_2))\beta \tag{5.1}$$

5.4 User Evaluation

We comparatively evaluated the prediction accuracy and the novelty of the resources recommended with ReDyAl with respect to three state-of-the-art recommendation algorithms relying exclusively on Linked Data to produce recommendations: dbrec [66], HyProximity transversal and HyProximity hierarchical [71]. This evaluation aimed to answer the following questions:

RQ1 *Which of the considered algorithms is more accurate?*

RQ2 *Which of the considered algorithms provides the highest number of novel recommendations?*

We decided to rely on a user study because we were interested in evaluating the novelty of proposed recommendations over the accuracy. Since we cannot expect that users rated all the items they already know, a user study can measure novelty more precisely than an offline study. On the other side, user studies are more expensive to conduct than offline studies, for this reason, we focus on recommendation algorithms based only on Linked Data, and we did not consider algorithms which exploit traditional techniques or combine Linked Data with traditional techniques. We plan to conduct other experiments to compare our method with other methods and investigate the effectiveness of our approach combined with traditional approaches.

Although our algorithm is not bound to any particular dataset, we applied it to DBpedia because it is a general dataset that offers the possibility to evaluate the results in different scenarios. DBpedia is one of the biggest datasets in the Web of Data and the most interlinked [69]. Furthermore, it is frequently updated and continuously grows.

5.4.1 Experiment

A user study was conducted involving 109 participants. The participants were mainly students of Politecnico di Torino (Italy) and University of Cauca (Colombia) enrolled in IT courses. The average age of the participants was 24 years old, and they were 91 males, 14 females, and 4 of them did not provide any information about their sex. Although the proposed algorithm is not bound to any particular domain, this evaluation focused on movies because we aimed at applying our algorithm in the mobile application presented in Section 5.5.1 (which suggest movies) and in this domain a quite large amount of data is available on DBpedia. Additionally, it was easier to find participants, since no specific skills are required to express an opinion about movies. The algorithms were compared within subjects [22] since each participant evaluated recommendations from different algorithms, as it is explained in the following.

The evaluation was conducted as follows. A list of 20 recommendations generated from a given initial movie was presented to the participants. For each recommendation two questions were asked:

Q1 *Did you already know this recommendation? Possible answers were: yes, yes but I haven't seen it (if it is a movie) and no.*

Q2 *Is it related to the movie you have chosen? Possible answers were: I strongly agree, I agree, I don't know, I disagree, I strongly disagree. Each answer was assigned a score from 5 to 1 respectively.*

We developed a website[4] to collect the answers from the participants. The participants were able to choose an initial movie from a list of 45 movies selected from the IMDB top 250 list.[5] The first 50 movies were considered, and five movies were excluded because they were not available in DBpedia. Choosing these movies ensured participants to know them, but was also a limitation: the corresponding DBpedia resources are very well linked. Thus we could not properly evaluate the algorithm on poorly linked initial movies. The movies were presented to the user in a random order to avoid having most of the participants evaluating recommendations for the same initial movies (e.g. the first in the lists). When a participant selected an

[4] http://natasha.polito.it/RSEvaluation/
[5] http://www.imdb.com/chart/top

initial movie the tool provided the corresponding list of recommendations with the questions mentioned above. The recommendations were presented in a randomized order. Each participant was able to evaluate recommendations from as many initial movies as he wanted, but he had to answer the questions for all the recommendations, i.e. was not possible to answer only to part of the questions for the initial movie chosen. As a result, the recommendations of the lists for 40 out of 45 initial movies were evaluated by at least one participant and each movie was evaluated by an average of 6.18 participants. The dataset with the initial movies and the lists of recommendations is available online.[6]

Each list of 20 recommendations was pre-computed. In particular, recommendations were generated for each of the 45 initial movies with each of the four different algorithms. Then, the recommendations generated by each algorithm were merged in a list of 20 recommendations to be shown to the participants. To do this, we created a list of 40 recommendations by selecting the first 10 pre-computed recommendations for each algorithm, and we ordered them by the similarity computed by each algorithm since each algorithm ranks its recommendations by using its semantic similarity function with values between 0 and 1. Then we eliminated eventual duplicates since more than one algorithm could provide the same recommendation. The final list was obtained considering the first 20 recommendations of the merged list.

With regard to the questions stated at the beginning of this section, to answer RQ1, the Root Mean Squared Error (RMSE) [22] was computed, and to answer RQ2 the ratio between the number of evaluations was computed in which the recommended item was not known by the participants and the total number of evaluations. For the RMSE measure, scores given by the participants when answering to Q2 were considered as reference and were normalized in the interval [0, 1], and these scores were compared to the similarities computed by each algorithm since each algorithm ranks its recommendations by using its semantic similarity function.

5.4.2 Results

The results of the evaluation are summarized in Figure 5.1, which compares the algorithms on their RMSE and novelty. The “sweet spot” area represents the conditions in which an algorithm has a good trade-off between novelty and prediction

[6] http://natasha.polito.it/RSEvaluation/faces/resultsdownload.xhtml

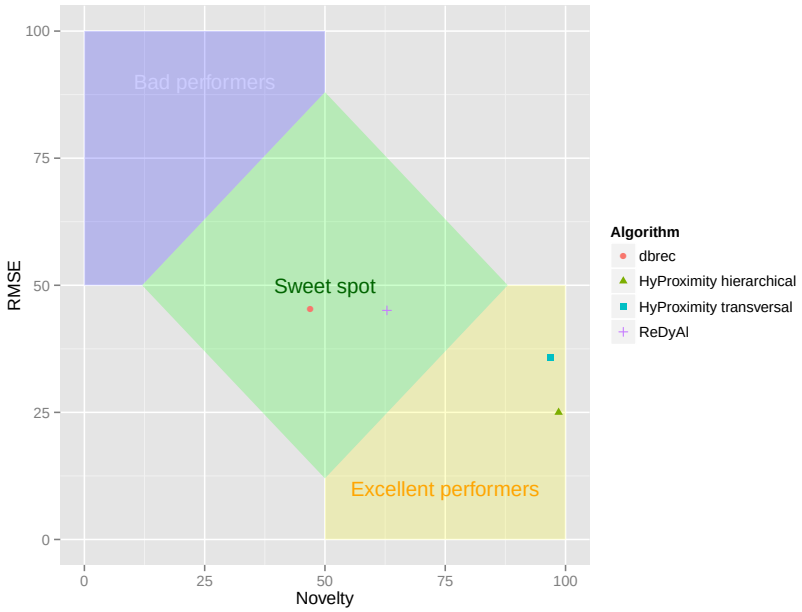


Fig. 5.1 Prediction accuracy and novelty of the algorithms evaluated.

accuracy. In effect, presenting a high number of recommendations not known to the user is not necessarily good because it may prevent him to assess the quality of the recommendations: for example having in the provided recommendation a movie which he has seen and which he liked may increase the trust of the user in the RS.

Regarding RQ1, HyProximity accounts for the lowest RMSE measures (with 25% and about 36% for the hierarchical and transversal versions respectively), but these results are less significant due to the low number of answers to Q2 for these algorithms. In effect, this means that the RMSE was computed over a low number of recommendations. For both ReDyAl and dbrec the RMSE is roughly 45%. Concerning RQ2, the two versions of HyProximity account for the highest values (hierarchical approximately 99%, while transversal about 97%). However, such a high rate of novel recommendations may confuse the user and prevent him from judging recommendations, as we have already explained. ReDyAl has a larger proportion of novel recommendations than dbrec. These two algorithms account respectively for about 60% and 45%.

The recommendations generated by HyProximity in both transversal and hierarchical version collected a low number of answers to Q2 because most of the recommendations generated by these algorithms were unknown as illustrated in

Table 5.1. Consequently, the RMSE was computed over a low number of recommendations. Thus, the results of these two algorithms related to RQ1 are less definitive than for the others, since for measuring the prediction accuracy only the evaluations for which the answer to Q1 was either *yes* or *yes but I haven't seen it (if it is a movie)* were considered.

Algorithm	**Yes**	**Yes but I haven't seen it**	**No**
ReDyAl	27.95	9.17	62.88
dbrec	41.10	11.95	46.95
HyProximity hierarchical	1.08	0.36	98.56
HyProximity transversal	1.32	1.89	96.79

Table 5.1 Percentage of answers to Q1 by algorithm.

We computed the Fleiss' kappa measure [86] for assessing the agreement of the participants in answering Q2. We considered the recommendations and in particular we considered as different the same recommendation when related to a different initial movie (i.e. when appearing in different lists of recommendations). We excluded recommendations not evaluated or evaluated by only one participant. The Fleiss' kappa is 0.79; according to Landis and Koch [87], this corresponds to a substantial agreement.

In conclusion, Figure 5.1 illustrates that ReDyAl and dbrec provides a good trade-off between prediction accuracy and novelty (sweet spot area), although ReDyAl performs better in novelty. HyProximity hierarchical and HyProximity transversal seem to be excellent performers since the RMSE is small and the novelty is high, but the RMSE was computed on few evaluations. An additional analysis of these two algorithms is needed to verify if the user can benefit from such a high novelty and if novel recommendations are relevant. In addition, further investigation is needed on poorly linked resources, since the choice of the initial movies focused on selecting popular movies to make easier the evaluation from participants, but the related resources were well linked. On poorly linked resources, we expect ReDyAl and HyProximity hierarchical keeping good recommendations since they can rely on categories, while dbrec and HyProximity transversal are likely to provide much fewer recommendations since they rely on direct links between resources.

5.5 Applications

In this section, we describe two real-life applications of our recommendation algorithm: a mobile application to suggest movies and an eTourism platform to recommend tourist attractions to visit.

5.5.1 Mobile Movie Recommendations

An implementation of ReDyAl has been integrated into a mobile application developed in collaboration with Telecom Italia (the major network operator in Italy). This application recommends movies based on DBpedia: when the user enters the title of a movie, the application provides the Wikipedia categories to which the initial movie is related to. In this way, the user may focus on a specific scope and can receive recommendations of related resources for any category. In addition, it is possible to view any recommendation to obtain additional information.

Our algorithm can provide cross-domain recommendations because it is independent of the domain and is applied on DBpedia, which is a general dataset. Thus, the recommended resources can be movies but also other relevant entities such as actors, directors, places of recording, books on which the movie is inspired, etc. Other advantages of using DBpedia as dataset are the high number of resources that it represents, the variety of domains addressed and the continuous update and growth since it is extracted from Wikipedia.

For example, given *The Matrix* as initial movie the categories which it belongs to are presented. The user may be more interested in martial arts, post-apocalyptic movies or he may prefer to consider all the movies from American directors; thus he can choose a category accordingly. By selecting Post-apocalyptic films, some resources are recommended. For each recommendation, it is possible to open a detailed view, which contains three tabs: the first provides a brief textual description, the second presents a graph view of the resource to show the main properties, and the third summarizes the main information in a tabular form. The graph view is illustrated in Figure 5.2. The graph is paginated, and few properties per page are presented to avoid information overload since the resource can have a very high number of properties. This view can be useful also to explain the recommendation: for instance, the user can understand that the recommended resource has the same

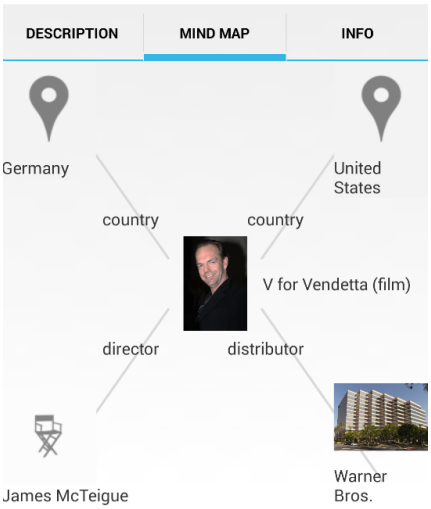


Fig. 5.2 The graph view of *V for Vendetta*.

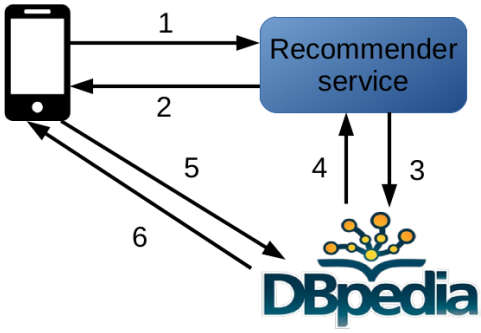


Fig. 5.3 The interactions between the main modules of the application.

director or the same leading actor as the initial movie. The graph view is based on DBpedia Mobile Explorer, a Linked Data visualization framework for the mobile environment, which enables the application to hide the underlying complexity of the Linked Data to the users by processing the resources to be presented received from DBpedia. The framework is presented in Section 7.3.

The application is based on a client-server architecture, and the main modules are DBpedia, a RESTful recommender service[7] which exposes our algorithm, and the mobile user interface. The main flow of interactions is represented in Figure 5.3. The mobile application asks for recommendations specifying an initial resource and optionally a scope such as a Wikipedia category (1). The recommender service answers with a list of scopes if no scope was provided or with a list of recommendations in the scope specified, otherwise (2). The recommender service relies on DBpedia to provide recommendations (3, 4) and the mobile application retrieves the resources to be visualized from the dataset (5, 6). The recommender service is developed in Java, while the client is an Android mobile application. The two modules use JSON as data-interchange format, while the mobile application retrieves resources from DBpedia serialized in JSON-LD.[8] The mobile application is going to

[7] http://natasha.polito.it/LDRecommenderWeb/
[8] http://json-ld.org/

be published on Google Play, but the Android Package (APK) of the first version is already available on the Web.[9]

5.5.2 eTourism Platform

Due to the increase of Linked Data published on the Web, it is more likely to find information related to real life concepts and increase the information associated with them in the form of User Generated Content (UGC) using Semantic Annotation techniques. In particular, we consider a tourism scenario, where more detailed information about attractions of a city and other POIs (like monuments, hotels, etc.) can be extracted from datasets in the Web of Data. For example, in DBpedia, we have information about many tourist attractions, such as the Colosseum, or St. Peter's Basilica in Rome. Thus, when a mobile user is in a tourism situation like visiting a city, the processing of the information associated tourist attractions of a city or other POIs could enable more reliable recommendations for more interesting places. This use case is detailed in Subsection *Use Case*.

This is possible because any resource in Web of Data has a URI, i.e. it is uniquely identified. Hence, it is possible to access it to get the information about the object it represents. Nowadays it is also simpler to publish information on the Web. Any user can easily insert new content in textual or multimedia format (with the use of contemporary mobile devices, it is becoming more and more real time), e.g. probably it is possible that some news appears on Twitter before than on any news portal. For example, considering a car accident in a city, witnesses can post information or content about it even before the local press agency arrives. Last but not least, it is also easier to link information with already existing resources. In fact, much effort has been done by the Linked Data community to provide more ways to increase publication of Linked Data. There are plenty of tools to annotate raw data and link them, e.g. DBpedia Spotlight [88] or the FI-WARE Semantic Annotator GE (for which details are provided in Subsection *Semantic Annotator*).

Another element to consider is that the structure of Linked Data can be exploited. In a tourism use case, it may be useful consider only hotels or only parks within a city. This can be done since Linked Data is structured: for example

[9] https://www.dropbox.com/sh/0q8d2mcbko9e2oj/AAASh-YHGz0MmG_Z8hH6mfWOa?dl=0

considering DBpedia, any resource belongs to one or more categories and categories of hotels and monuments exist. In addition, categories are organized in a tree hierarchy; thus it is possible to consider more general or more specific categories. If we refer to Saint Petersburg, then we can also consider the hotels (associated to `Category:Hotels_in_Saint_Petersburg`), monuments (corresnding to `Category:Monuments_and_memorials_in_Saint_Petersburg`) and so on. The whole city may be mapped to `Category:Buildings_and_structures_in_Saint_Petersburg`. Finally, information in Linked Data and its structure can be used to increase the user experience.

In the following, we propose a platform to support tourism use cases, such as the one described in Subsection *Use Case*, which exploit ReDyAl to provide recommendations based on Linked Data. The overall system is shown in Figure 5.4. Our platform communicates with mobile devices and with the Web of Data using web interactions, and it is made up of three main modules: a recommender system that implements ReDyAl, a UGC manager, and a semantic annotator. The first can provide information from Web of Data and UGC stored, while the second stores and retrieves UGC, which is also linked to resources belonging to the Web of Data using the annotator. In the following, we provide a description of each component of the platform (apart from the recommender system which is based on ReDyAl, which has already been presented in Section 5.3).

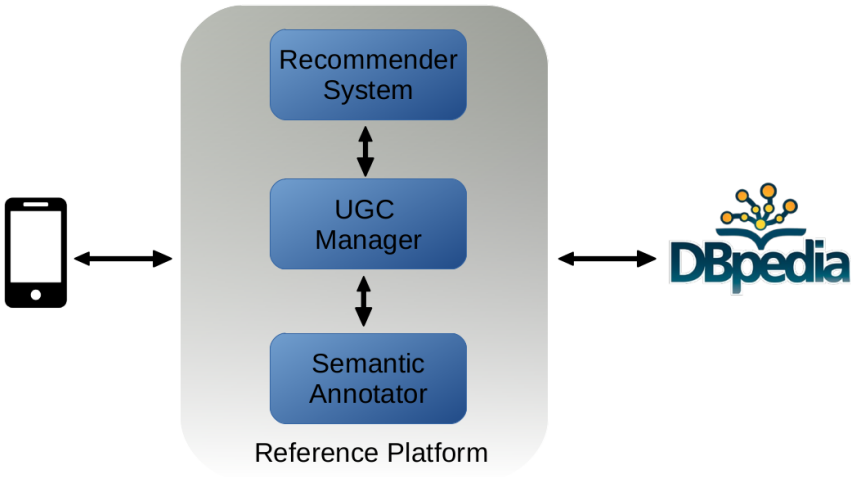


Fig. 5.4 The overall system architecture.

Semantic Annotator

Thanks to the FI-WARE EU project,[10] it is possible to reuse existing components for creating modern applications and services. In this case, we have reused the Semantic Annotator GE, which is publicly available from the FI-WARE catalogue.[11] The decision of reusing an existing component or creating a new one was taken after careful analysis, in which it was concluded that the component that provides the service offers greater advantages and reduces the time of integration on the platform. Besides, it encourages the standardization of the existing modules.

To extract semantic information, each plain text information associated with received content is analyzed by the Semantic Annotator GE (which architecture is shown in Figure 5.5). First, the Text Processor module identifies the source language; then a morphological analysis is performed using FreeLing[12] configured for the identified language. From this analysis, proper nouns lemmas are extracted while other part-of-speech are discarded. At this time, non-numeric proper nouns lemmas with a score of at least 0.2 are preserved and merged with plain text tags to compute a well-defined list of unique (multi) words. At this stage, the module uses term frequency to process the title further and to extract other potential relevant words.

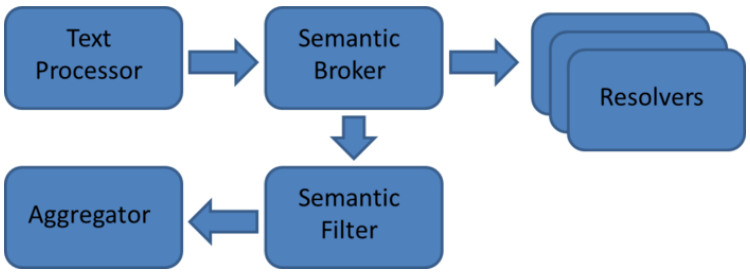


Fig. 5.5 Semantic Annotator GE at a glance.

The next step involves the Semantic Broker, which is assisted by a set of resolvers that perform full-text or term-based analysis based on the previous output. Such resolvers are aimed at providing candidate semantic concepts referring to Linked Data as well as additional related information if it is available. Resolvers may be domain or language specific, or general purpose. For term-based analysis, each word of the previously computed list is individually processed to identify a list of

[10] https://www.fiware.org/
[11] http://catalogue.fi-ware.org/enablers/semantic-annotation
[12] http://nlp.lsi.upc.edu/freeling/

candidate Linked Data resources to match with. A set of predefined services, such as DBpedia, Sindice[13] and Evri[14] are invoked in parallel.

The Semantic Filtering module processes candidate Linked Data resources received by the broker and performs a disambiguation based on the DBpedia score and the string similarity between each surface form and its corresponding list of candidates, which relies on the Jaro-Winkler distance [89]. This function aims at maximizing both values to identify the "preferred" candidate. In this process, after several empirical tests, candidates with distance lower than 0.8 are discarded at this stage, unless their DBpedia score is the maximum. Automatic annotation is performed using the "preferred" candidate identified during this step.

UGC Management

This layer of the platform is responsible for saving UGC and its semantically enriched version. Thus, for each given UGC entry, we have:

- UGC itself (any given multimedia file);
- Original associated plain information (stored in a SQL database);
- Associated semantic information represented in RDF (stored in a triple store).

Additionally, this layer offers the means to retrieve specific content, either through the invocation of a REST API (to get the multimedia contents attached) or performing more complex SPARQL queries in the public endpoint.

Use Case

The previously described platform can be applied in a tourism scenario and can be exploited by an application that assists tourists by providing them suggestions about places to visit, accommodations, and other points of interest (POIs). Furthermore, it allows users to share their experience by providing content such as pictures, videos, reviews, and comments about places they have been. Then, this content is available

[13] http://sindice.com
[14] http://www.evri.com

to other users and can be exploited to enrich information about tourist attractions and other facilities.

For example, if a user is in Saint Petersburg, some tourist attractions are the Peter and Paul Fortress, the Saint Isaac's Cathedral and the Hermitage Museum. All of these tourist attractions have a corresponding representation in DBpedia; thus they support user interactions, since users can decide if visit them or not by consulting the information provided by DBpedia. There are two possible interactions:

1. user device receives information about nearby tourist attractions;
2. user publishes information regarding the tourist attractions from his device (generates content).

The first enables users to be provided with recommendations about tourist attractions, while the latter allows users to share their experiences and increase the amount of available information to other tourists.

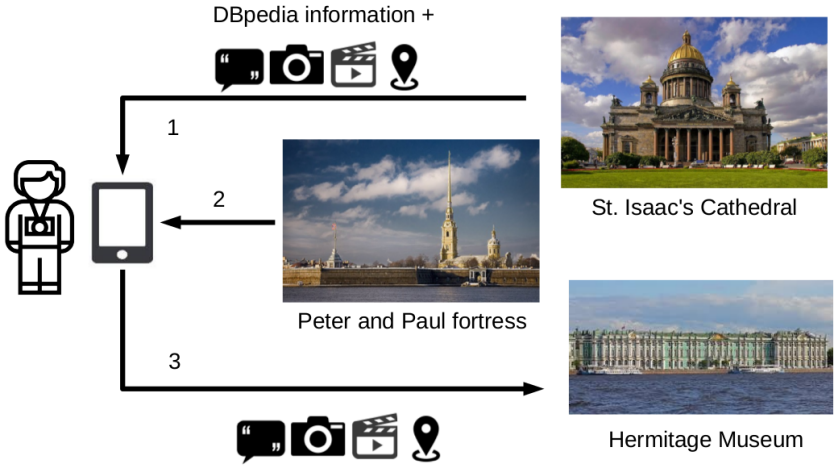


Fig. 5.6 Possible interactions in a eTourism use case.

Figure 5.6 illustrates the two different possible interactions for a user in Saint Petersburg. He is at the Hermitage Museum, and his device receives recommendations about the Peter and Paul Fortress and the Saint Isaac's Cathedral, which are nearby. He can access the DBpedia information related to each of them and also the related UGC shared by other users (1, 2). Then he decides to take a picture with his friends and make it available to other tourists; thus he adds information to the Hermitage museum by using the semantic annotator and UGC manager (3).

This use case has been developed together with Telecom Italia, the primary network provider in Italy, and it shows how information of real word objects stored in the Web of Data can be exploited in a tourism scenario. In this case, information about tourist attractions and POIs allows users to decide how to behave, e.g. if visiting or not a particular monument.

5.6 Conclusions and Future Work

We presented ReDyAl which is a hybrid algorithm that dynamically uses both the transversal and hierarchical approach for discovering resources. It is independent of the application domain and, although we applied it to DBpedia, it could be easily adapted to other datasets in the Web of Data. It relies only on Linked Data and does not require to reduce the set of resources and links of the dataset to those belonging to a particular domain.

We evaluated and compared our algorithm against three state-of-the-art algorithms by conducting a user study and we also showed two practical applications of the algorithm by presenting a mobile application that provides movie recommendations and an eTourism mobile application, both relies on DBpedia. Although the algorithm could be applied to other datasets in the Web of Data, we selected DBpedia because it is a general dataset; thus cross-domain recommendations were possible. Besides, there is a high number of resources represented, a variety of domains addressed and it is continuously updated since it is extracted from Wikipedia. The user study demonstrated that ReDyAl improves in the novelty of the results discovered, although the accuracy of the algorithm is not the highest (due to its inherent complexity). Although ReDyAl is not bound to any particular domain, the study focused on movies because Telecom Italia was interested in a related use case. Furthermore, in this domain there is a quite large amount of data available on DBpedia and participants were not required to have specific skills.

Future work includes studying the relevance under different domains and improving the accuracy of ReDyAl while maintaining its novelty. We plan to conduct other studies to compare it with traditional techniques and with approaches which combine Linked Data with traditional techniques. We are also working on combining ReDyAl with collaborative filtering techniques to take user preferences into account while providing recommendations. It is worth to note that ReDyAl could be extended to

consider more than one resource in input (e.g. all the resources rated positively by the user). In order to do this, ReDyAl could be executed multiple times to generate recommendations given a number of initial resources, and subsequently, the results could be merged. However, this would significantly increase the response time since the algorithm relies on SPARQL queries to discover candidate recommendations through the links among resources, which is computationally expensive. Thus, we should study how to do this taking performance into account. Another resource to consider could be the current context of the user. Context-awareness is addressed in Chapter 6. In particular, in that chapter, we present a context-aware recommendation technique. We could also extend ReDyAl with the context-aware recommendation method presented in that chapter. For example, that method could be used to select an initial resource for ReDyAl from a set of user ratings based on the context.

Chapter 6

Leveraging Ontologies for Context-Aware Recommendations

6.1 Introduction

Context-Aware Recommender Systems (CARS) are a particular category of recommender systems which exploits contextual information to provide more useful recommendations. For example, in a temporal context, vacation recommendations in winter should be very different from those provided in summer. Similarly, a restaurant recommendation for a Saturday evening with your friends should be distinct from that suggested for a workday lunch with co-workers [1].

Nowadays contextual information such as time and location are easy to be obtained with modern devices. However, also other parameters may be considered, such as the company (alone, with friends, with the one's partner) which may be relevant when recommending movies or vacations. In addition, the exact context sometimes can be too narrow, as Adomavicius and Tuzhilin [14] exemplified by considering the context of watching a movie with a girlfriend in a movie theater on Saturday. Using this exact context may be problematic for several reasons. First, certain aspects of the overly specific context may not be significant. For example, a user which watch a movie with the one's partner in a theater may have the same preferences on Saturday and Sunday, but they may change on Wednesday. Therefore, it may be more appropriate to use a more general context specification, i.e. weekend instead of Saturday. Second, the exact context may not have enough data for accurate

rating prediction, which is known as the data sparsity problem. Thus it may be useful to refer to a more general context such as watching a movie with the one's partner in a movie theater on the weekend, watching a movie with someone in a movie theater on the weekend, and so on.

Additionally, often user preferences and items representation depend on the application domain addressed or on the particular recommendation approach used. Thus, a significant effort is required to adapt the recommender system to another domain or to change the approach to use.

In this chapter, we address the problems previously mentioned and we focus on the following research questions:

- *Is it possible to represent context by combining different dimensions (such as time, location, mood, etc.) and representing different granularities for each dimension (e.g. the precise time moment, the day of the week or the season)?*
- *Is it possible to represent user preferences and items in such a way that can be adapted to different application domains and combined with different recommendation approaches?*

We distinguish three forms of context-aware recommendation process: *contextual pre-filtering*, *contextual post-filtering*, and *contextual modeling* [14]. *Pre-filtering* approaches use the current context to select a relevant subset of data on which recommendation algorithm is applied. *Post-filtering* methods exploit contextual information to select only relevant recommendations returned by some algorithm. *Contextual modeling* differs from other techniques as it incorporates the context into recommendation algorithm. We opted for a pre-filtering strategy because it can be used with existing recommendation algorithms and avoids an expensive search of an effective post-filtering approach, as explained in Section 6.2.[1]

We propose a new contextual pre-filtering approach which is based on two ontologies to represent context and user preferences: Recommender System Context (RSCtx)[2] which describes the context, and Contextual Ontological User Profile (COUP), which represents user preferences. COUP is based on Structured-Interpretation Model (SIM) [90] and consists of multiple ontological modules. We

[1] An exhaustive review of CARS is out of the scope of this thesis. The reader may refer to the survey of Adomavicius and Tuzhilin [14].

[2] http://softeng.polito.it/rsctx/

evaluated our approach through an offline study with a rating prediction task which showed that the usage of the proposed ontologies and our pre-filtering technique with a number of well-known recommendation algorithms significantly improves the accuracy of prediction according to the Mean Absolute Error (MAE) measure.

The rest of the chapter is organized as follows: Section 6.2 provides an overview of CARS, Section 6.3 presents related work, Section 6.4 introduces our ontology to represent the context, while Section 6.5 addresses the overall recommendation approach and the representation of user preferences. We detail the evaluation process and its results in Section 6.6 and we conclude in Section 6.7.

6.2 Context-Aware Recommender Systems

As explained in Section 2.2, traditionally RS deal with users and items. Thus, the function to estimate users' rating is two-dimensional. On the contrary, CARS need to incorporate available contextual information into the recommendation process as an additional type of data. The user's preferences are usually expressed as ratings and are modeled as the function of not only items and users, but also of the context. This implies to define ratings with a three-dimensional rating function $f : U \times I \times C \rightarrow R$, where R is the domain of ratings and a totally ordered set, U and I are the domains of users and items respectively, and C specifies the contextual information associated with the application [14].

Adomavicius and Tuzhilin [14] distinguished two approaches to exploit contextual information in RS: recommendation via context-driven querying and search, and recommendation via contextual preference elicitation and estimation. Systems which rely on the former strategy typically use contextual information to query or search a certain repository of resources and suggest the best matching resources. The context is obtained either directly from the user, e.g., by specifying current mood or interest, or from the environment, e.g., acquiring local time, weather, or current location. However, recently the trend is exploiting the latter method. Techniques that adopt it model and learn user preferences, e.g., by observing the interactions with the system or by obtaining preference feedback from the user on previously recommended items. To model users' context-sensitive preferences and generate recommendations, these techniques typically either adopt existing collaborative-filtering, content-based,

or hybrid recommendation methods to context-aware recommendation settings, or apply data analysis techniques from data mining or machine learning.

In general, we can identify three main components of the recommendation process: the input data, the recommendation function and the recommendation list. We can apply contextual information at several stages of this process and distinguish three types of CARS based on which component the context is used in [14]:

Contextual pre-filtering is the contextualization of the recommendation input. The specific context considered drives data selection or data construction. The current context is used for selecting or constructing the relevant set of ratings. Then, ratings can be predicted using any traditional two-dimensional recommender system. This method can be applied on existing recommendation method [16]. This is its primary advantage. The disadvantage is that context can be too narrow as explained in Section 6.1.

Contextual post-filtering is contextualization of the recommendation output. The contextual information is initially ignored, and the ratings are predicted using any traditional recommender system on the entire data. Then, the resulting set of recommendations is adjusted for each user using the current context. The adjustment can be a filter or a modification of the recommendation list. In the former case, some items are removed because they are irrelevant for the current context, while in the latter the ranking of items varies based on the context. Similarly to pre-filtering, it can be applied to available recommendation approaches and it needs a properly generalized context.

Contextual modeling is the contextualization of recommendation function. The context is used directly in the modeling technique as part of rating estimation. This method requires a multidimensional recommendation function. Thus it is not possible to reuse existing recommendation techniques, although some have been extended to manage more dimensions.

Panniello et al. [91] compared pre-filtering and post-filtering approaches and showed that the best approach depends on the application. They suggest exploiting pre-filtering when it performs better than the un-contextual case because it avoids an expensive search of an effective post-filtering method.

6.3 Related Work

We distinguish between works which addressed representation of context and other ontology-based recommender systems proposed. The former are presented in Section 6.3.1 while the latter are briefly described in Section 6.3.2.

6.3.1 Context Representation

In this section, firstly we address ontology-based context modeling and then we review context representation for recommender systems.

Many context ontologies have been proposed in the context awareness community. There are a number of surveys which review the literature relevant to context modeling in general [92, 93], or focus on ontology-based models [94, 95]. In addition, Costabello [96] presented and compared a number of ontology-based context models against a set of requirements. These requirements also fit our purpose, therefore in the following, we present the requirements and summarize Costabello's comparison, obviously also considering RSCtx. The relevant context aware and ontology engineering requirements are:

R1. Domain independence. Some context ontologies have been created to model a given domain-specific scenario. Others adopt a domain-independent approach.

R2. Coverage. The ontology must guarantee a proper level of completeness for the desired contextual dimensions. The model must support multiple context dimensions such as device features, user preferences, location and time.

R3. Variable Context Granularity. Certain ontologies model context dimensions at different levels of granularity. For example, the location might be expressed in terms of latitude and longitude, or with a label assigned to a place (e.g. office, beach, cinema, etc.).

R4. Core ontology approach. The vocabulary must adopt a modular design, thus focusing on modeling core classes and properties that will be extended by third-party domain specialists.

Costabello [96] also considered some requirements related to the Linked Data principles, which also fit our purpose:

R5. Open World Assumption. The Web of Data is an open environment, and describing context in this scenario must consider third-party extensions unknown beforehand. Extensibility must be obtained with little effort; thus add-ons must not impact on the already existing model.

R6. Lightweight Ontology. According to Linked Data best practices [35], the goal is to keep ontologies small and straightforward.

R7. Reuse of Existing Terms. Linked Data best practices favor the reuse and the combination of classes and properties of existing vocabularies. This is done to prevent the proliferation of terms and reduce the range of choices when modeling data.

R8. Availability on the Web. Web of Data vocabularies are published on the Web, and accessible according to Web of Data best practices.[3] Moreover, they are associated with an HTML page, the "namespace document", whose task is to provide a textual description of the vocabulary rationale, along with classes and properties explanation and examples.

Work	R1	R2	R3	R4	R5	R6	R7	R8
PRISSMA[3] [96]	●	●	○	●	●	●	●	●
DCO[5]	●	○		●	○			●
SOUPA [97]	●	●		●	●		○	
CoOL [98]	●	○	○	●	○			
CONON [99]	●	●	●	●	●			
CoDaMos [100]	●	●			●			○
Korpipää et al. [101]	●	○			○			
Hervás and Bravo [102]	●	●		●				
RSCtx	●	●	●	●	●	●	●	●

Table 6.1 A comparison of ontology-based context models [96]. Full support is identified by ●, partial support by ○, no support by the empty cell.

Following these requirements, Costabello [96] compared a number of ontologies which modeled context and proposed PRISSMA,[4] a vocabulary designed to model client generated context data. In the following, we present the main features of this vocabulary and of the other related works shown in Table 6.1.[5] PRISSMA satisfies

[3] https://www.w3.org/TR/swbp-vocab-pub/
[4] http://ns.inria.fr/prissma
[5] An exhaustive review of the related literature is out of the scope of this thesis. The reader can refer to more complete surveys [92–95].

most of the requirements mentioned above, although variable context granularity is only partially satisfied. All the works provide coverage and are domain independent, and all but one support (at least partially) the open world assumption. The only other ontology published on the Web is the Delivery Context Ontology (DCO),[6] a modular and fine-grained vocabulary to model mobile platforms. It does not provide linking with other vocabularies, and it is not considered a lightweight ontology. The SOUPA ontology [97] is an OWL ontology which is extensible, i.e. supports the open world assumption and reuses external ontologies, but it does not comply with Linked Data principles, for example, it is not publicly available on the Web. CoOL [98] is a modular OWL ontology, which is grounded on F-Logic and uses features typically avoided in a lightweight ontology. CONON [99] is another modular OWL ontology, which is not published on the Web and does not reuse existing vocabularies. CoDaMos [100] is an extensible OWL ontology that is available on the Web but no namespace vocabulary is present. It is not lightweight and does not reuse other vocabularies. Korpipää et al. [101] presented a context model designed for mobile, context-aware applications. This model is general but does not reuse existing terms, and it is not extensible. Hervás and Bravo [102] proposed a modular context model composed by independent ontologies. It supports extensions, although it does not reuse already existing linked data ontologies.

Context Representation for Recommender Systems

Various works address context representation for RS. Abowd et al. [103] distinguished between primary and secondary context: the former can be directly measured, while the latter needs to be derived from other types of contextual information.

Kaminskas and Ricci [104] reviewed the literature about contextual music retrieval. They distinguished between environmental, user-related and multimedia context. The first refers to information about the location of the user, the current time, weather, temperature, etc.. The second concerns information about the activity of the user, the user's demographic information, and the emotional state. The third applies to other types of information the user is exposed to besides music, e.g., text and images. In addition to traditional dimensions (time location etc.) the authors suggested traffic, noise and light level. As multimedia context, they mention text and

[6] http://www.w3.org/TR/2009/WD-dcontology-20090616/

images. They indicated some cases in which it can be useful to consider this kind of context, e.g. for adapting music to text context as done by Cai et al. [105].

Baltraunas et al. [106] proposed an approach to assess which contextual factors are important and to which degree they influence user ratings. They conducted a study in which users were asked to judge whether a contextual factor affects the rating given a certain contextual condition. In their survey they focus on tourism domain and consider budget, time availability, and transport in addition to traditional dimensions. RSCtx supports most of the addressed dimensions and distinguishes between user-related and environmental context. It does not address multimedia context, but it considers the device features.

6.3.2 Ontology Based Recommender Systems

It has been proved that the ontological user profile improves recommendation accuracy and diversity [107]. More specifically, a number of ontology-based and context-aware RS have been proposed. In the following, we briefly review the main approaches, but a more detailed description of ontology-based techniques is provided by Di Noia and Ostuni [10] and Lops et al. [25].

AMAYA allows management of contextual preferences and contextual recommendations [108]. AMAYA also uses an ontology-based content categorization scheme to map user preferences to entities to recommend. News@hand [109] is a hybrid personalized and context-aware recommender system, which retrieves news via RSS feed and annotates by using system domain ontologies. User context is represented by a weighted set of classes from the domain ontology. Rodriguez et al. [110] proposed a CARS which recommends Web services. They use a multi-dimensional ontology model to describe Web services, a user context, and an application domain. The multi-dimensional ontology model is made up of three independent ontologies: a user context ontology, a web service ontology, and an application domain ontology, which are combined into one ontology by some properties between classes from different ontologies. The recommendation process assigns a weight to the items based on a list of interests in the user ontology. All these works focus on a particular domain and an ad-hoc algorithm, while our approach to represent user preferences is cross-domain and can be applied to different recommendation algorithms.

Hawalah and Fasli [111] suggest that each context dimension should be described by its own taxonomy. Time, date, location, and device are considered as default context parameters in the movie domain. It is possible to add other domain specific context variables as long as they have a clear hierarchical representation. Besides context taxonomies, this approach uses a reference ontology to build contextual personalized ontological profiles. The key feature of this profile is the possibility of assigning user interests in groups, if these interests are directly associated with each other by a direct relation, sharing the same superclass, or sharing the same property.

Other works use ontologies and taxonomies to improve the quality of recommendations. Middleton et al. [5] used an ontological user profile to recommend research papers. Both research papers and user profiles are represented through a taxonomy of topics, and recommendations are generated considering topics of interest for the user and papers classified in those topics. Mobasher et al. [6] proposed a measure which combines semantic knowledge about items and user-item rating, while Anand et al. [2] inferred user preferences from rating data using an item ontology. Their approach recommends the items using the ontology and inferred preferences while computing similarities.

6.4 The Recommender System Context Ontology

Recommender System Context (RSCtx) extends PRISSMA, a vocabulary based on Dey's definition of context [112]. PRISSMA relies on the W3C Model-Based User Interface Incubator Group proposal,[7] which describes mobile context as an encompassing term, defined as the sum of three different dimensions: user model and preferences, device features, and the environment in which the action is performed. A graph-based representation of PRISSMA is provided Figure 6.1.

We designed RSCtx following METHONTOLOGY [113], a well know ontology design method. We assumed there is a predefined set of contextual dimensions in a given application, each with a defined set of attributes and we modeled the contextual information relevant to provide recommendations. We did not focus on any particular domain, on the contrary, we aimed at reusing the ontology in different

[7] http://www.w3.org/2005/Incubator/model-based-ui/XGR-mbui/

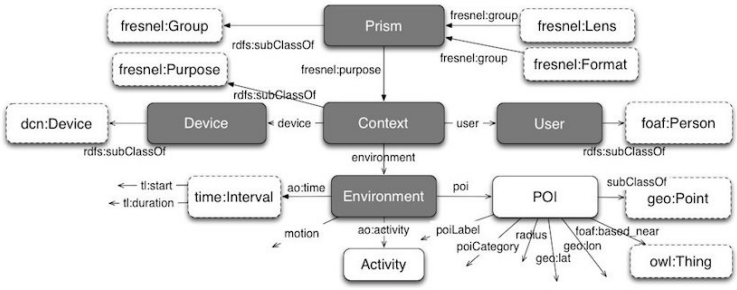


Fig. 6.1 The PRISSMA vocabulary [96].

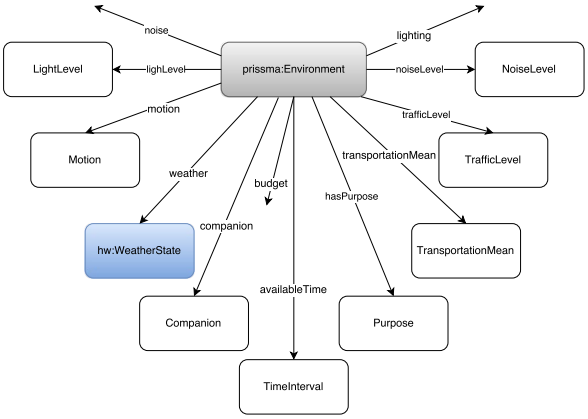


Fig. 6.2 The relations and concepts which extend `prissma:Environment`.

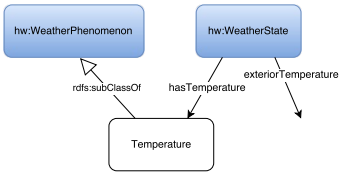


Fig. 6.3 Temperature representation in our ontology.

applications. As in PRISSMA, the point of view used to describe the context itself is the application point of view, we considered the user itself as part of the context.

We needed a more detailed representation of the environment, to consider other contextual dimensions such as the purpose of the user and the weather. Figure 6.2 shows how `prissma:Environment` has been extended, by adding a number of properties and related concepts. To represent the weather, we integrate `hw:WeatherState` from the Weather Ontology.[8] In this ontology the temperature is represented as the room temperature, thus we defined a new class to represent symbolic values of temperature (such as warm, cold, etc.) and an attribute to represent numeric values, as shown in Figure 6.3.

We also extended the time and location representations. We needed a more expressive model of these two dimensions, since asking for recommendations which

[8] https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/WeatherOntology.owl

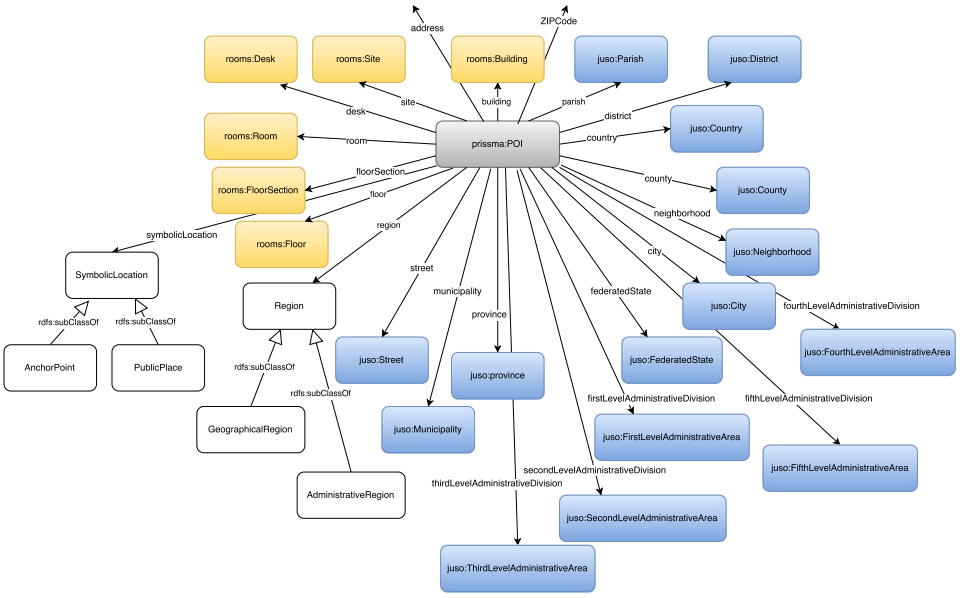


Fig. 6.4 The concepts and relations of RSCtx which represent the location dimension.

have the same time stamp and the coordinates of the actual context is too restrictive, and the recommender system may not have enough data. On the contrary, by generalizing the context (for example distinguishing among weekend and working day, or considering the city or neighborhood instead of the actual user position) may enable the recommender system to provide recommendations. The concept `prissma:POI` has been extended with various properties to represent the location in the context of a particular site by integrating the Buildings and Rooms vocabulary.[9] Furthermore, other properties related to the hierarchical organization of the location (such as the neighborhood, the city and the province of the current user position) have been added, and some concepts from the Juso ontology[10] have been reused. Figure 6.4 depicts relations and attributes which characterize a location. Yellow rectangles indicate concepts from rooms vocabulary, while blue ones are taken from Juso. The representation of time augments `time:Instant` defined in the Time ontology.[11] Some time intervals have been defined: the hours and the parts of the day (morning, afternoon, etc.). Besides, days of the week are classified in weekdays or weekend and seasons are represented. Figure 6.5 illustrates how time is represented and the relations with PRISSMA and the Time ontology.

[9] http://vocab.deri.ie/rooms
[10] rdfs.co/juso/latest/html
[11] https://www.w3.org/2006/time

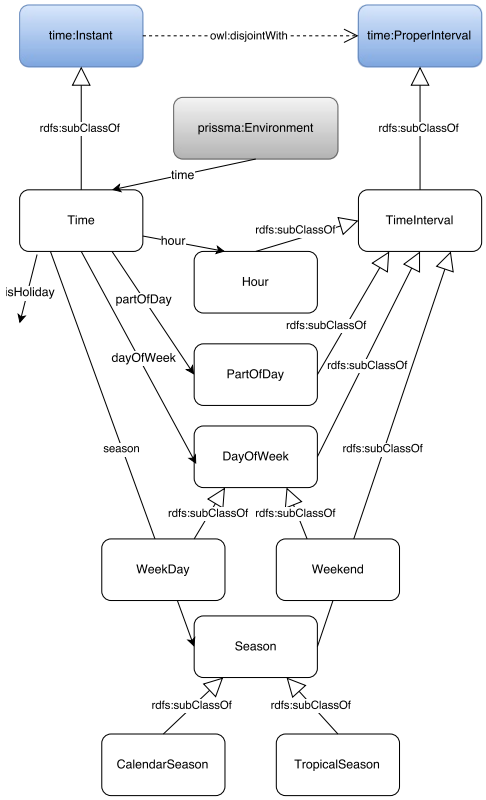


Fig. 6.5 Time representation in our ontology as extension of Time and PRISSMA ontologies.

Furthermore, we extended the user representation adding some dimensions which may be of interest, as the emotional, mental and physiological state of the user or his fitness. This can be interesting mainly in the medical or fitness domain, but the emotional state can affect the user also in taking other kinds of decisions, like choosing a movie to watch or music to listen to. Emotional, mental and physiological state concepts are equivalent to the emotional, mental and physiological state in the General User Model Ontology (GUMO) [114], an ontology to describe the user which is available on the Web, although it is not compliant with Linked Data principles since it has not a namespace assigned. In addition, the emotional state is an extension of `emoca:Emotion`, which is defined in the Emotion Ontology for Context Awareness (EmOCA).[12] We added some attributes to the physiological state and also defined an arousal relation which reuses `emoca:Arousal`. Figure 6.6 depicts the user representation in our ontology.

[12] http://ns.inria.fr/emoca/

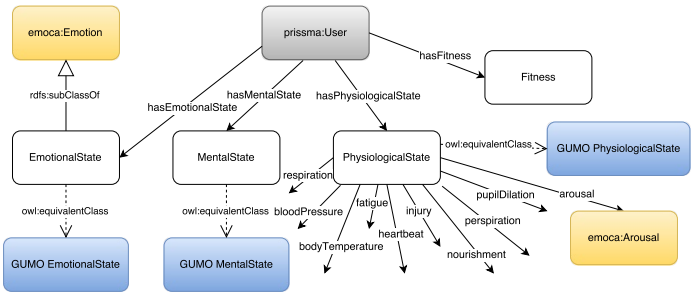


Fig. 6.6 User representation in RSCtx ontology.

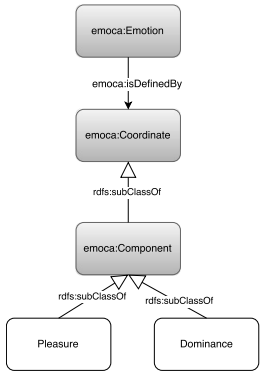


Fig. 6.7 Emotion model.

The emotion in EmOCA is represented according to Russel's model [115]. We extended `emoca:Emotion`, in particular we added pleasure and dominance as subclasses of `emoca:Component` in order to represent emotions according to the Pleasure Arousal Dominance (PAD) model [116] as well, as it is showed in Figure 6.7. In this way, we can indicate that the emotion is defined by valence and arousal using `emoca:isDefinedBy` to refer to Russel's model, while we can mean that the emotion is defined by pleasure, arousal, and dominance to refer to the PAD model. Furthermore, it is possible to relate to emotion just by indicating its category (such as joy, anger, disgust, etc.). In EmOCA, six categories have been already defined, which can also be used in RSCtx since the emotional state is a subclass of `emoca:Emotion`. We can add more categories in our ontology, although we have not done it.

6.5 Recommendation approach

6.5.1 The Contextual User Profile Ontology

To model user profiles we used the Structured Interpretation Model (SIM) [117, 118], which consists of two types of ontological modules, i.e. *context types* and *context instances*. Context types describe the terminological part of an ontology (TBox) and are arranged in a hierarchy of inheritance. Context instances describe assertional part of an ontology (ABox) and are connected with corresponding context types through a relation of instantiation. Context instances of more specific context types are linked to a context instance of a more general context type through a relation of aggregation. In the class hierarchy in a classical ontology there always exists a top

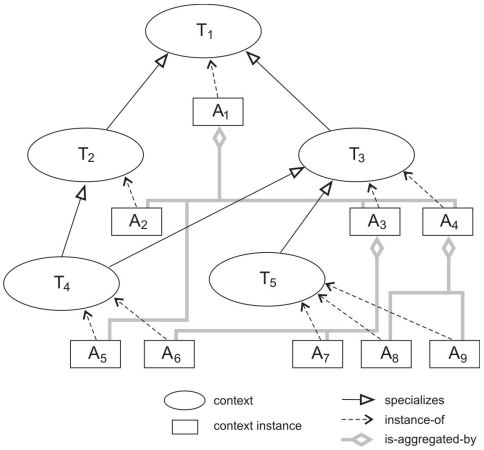


Fig. 6.8 SIM at a glance.

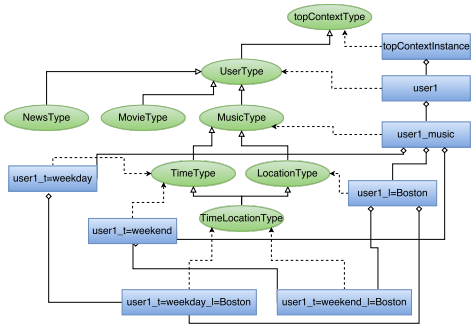


Fig. 6.9 An example of COUP.

concept, i.e. `Thing`. In a SIM ontology, there is a top context type and a top context instance connected by instantiation. It is possible to add multiple context instances to one context type and aggregate multiple context instances into one context instance. An overview of SIM is available in Figure 6.8.

The idea of adapting a SIM ontology as a user profile was proposed by Karpus and Goczyła [119]. They modeled contextual user profiles using only three context variables, i.e. location, time and mood, which influences a split of terminology into ontological modules. Our approach is different in some crucial aspects. First of all, we allow storage of many user profiles in the same SIM ontology. We also support a storage of preferences from multiple domains by adding context types related to different domains. Another difference is the number of context variables permitted. We add context types and context instances related to contextual parameters in a dynamic way. As a consequence, we can use as many variables as needed in our approach. An example of a contextual profile for one user is shown in Figure 6.9.

Only three modules in the example illustrated in Figure 6.9 are fixed: `UserType`, `topContextInstance` and `topContextType`. All others are configurable or can be added in a dynamic way. In `topContextType` we defined the concept `Rating` and its corresponding roles, e.g. `isRatedWith` and `hasValue`. `UserType` is artificial and is present in the SIM ontology because it enables to add many user profiles to the ontology. In the next level of the hierarchy, there are context types that describe domains of interests related to a recommender system which will use the profile. In the next levels, all context types and instances are added to the contextual user profile during the learning phase or later, when a new context situation occurs.

The general process of learning the user profile is as follows. At the beginning, there is just the RSCtx ontology and an empty contextual ontology, i.e. with the terminological part only. For a given user, an item is taken with the rating and the situation in which it was consumed from the user's history. The level of granularity of the context is checked with the RSCtx ontology and is changed if needed, e.g. shifting from *time = 2 p.m.* to *time = afternoon*. A context instance is created for this context if it is not already available. Finally, an item with its rating is added to the identified context instance. Each item is represented as a set of individuals of appropriate concepts defined in a domain context type.

6.5.2 Recommendation

We use the ontologies previously presented in a pre-filtering method integrated in the recommendation process. The aim is providing a context-aware technique to improve existing algorithms.

The system consists of three main functional modules: context detection and generalization, user profile and pre-filtering, and recommendation. In the first module, we used the RSCtx ontology to identify the user context from raw data and generalize it to the desired granularity level. The second module is responsible for building the user profile, finding a context instance that fits the actual user context, and returning only relevant user preferences. The last module uses well-known algorithms, e.g. Item *k*NN, User Average, SVD++, for providing recommendations. For this task we exploit implementations from the LibRec[13] library.

The general recommendation process is summarized in Figure 6.10. Given a user and his current situation, a proper generalization of his context is generated by using the RSCtx ontology. Then, an appropriate context instance from COUP is identified by using the generalized context. If a context instance is not found in the user profile, the generalization step is repeated to search for a module that corresponds to the new context. If it is found, user preferences are prepared to be used with a recommendation algorithm.

Given the current context, the generalization module creates a context instance in RSCtx and initialize all its properties. For example, given a time stamp the day of the week, and the part of the day can be set. Similarly, from latitude and

[13] http://www.librec.net/

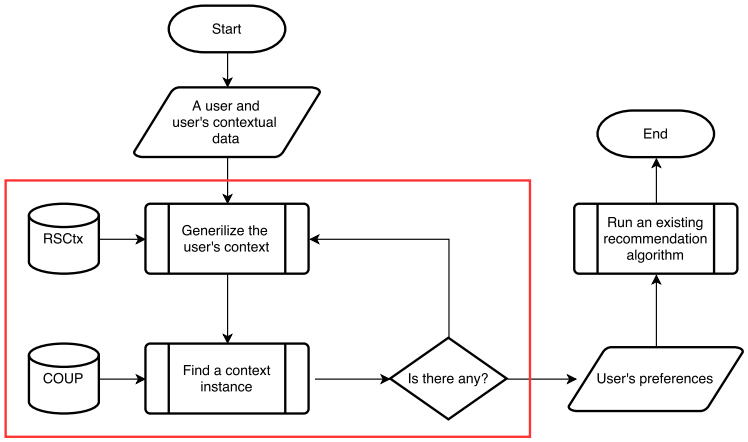


Fig. 6.10 General recommendation process.

longitude, it is possible to obtain the address, the city, and the country through a geocoding service. Once the instance is initialized, an initial granularity is set for each dimension, e.g. the day of the week for the time dimension and the city for the location dimension. The initial granularities to set are additional parameters of the generalization algorithm, since they may depend on the particular application scenario considered. It is possible to generalize the context by specifying a broader granularity for one or more dimensions. If a granularity is not given, the context is generalized of one step, e.g. switching from the part of the day to the day of the week. An example with the time dimension is showed in Figure 6.11. Given the time stamp corresponding to May, 5th 2017 at 14:30, the context is instantiated and the properties are initialized as depicted. The initial granularity is the day of the week, i.e. Friday in this case. If not enough preferences with time information equal to

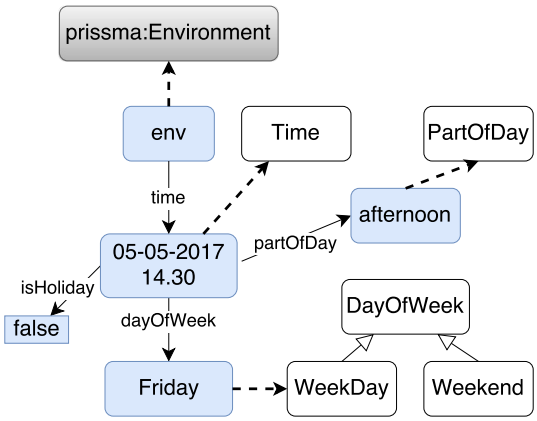


Fig. 6.11 A context instance in RSCtx with only the time dimension.

Friday are found in COUP, the context is generalized, for example instead of Friday, the granularity is set to weekday. The minimum number of preferences required is a parameter since it may depend on the recommendation algorithm used.

6.6 Evaluation

We conducted an offline study to evaluate the RSCtx ontology and the COUP ontology. We selected a number of algorithms, and we compared the accuracy of each algorithm when used as is and when combined with the proposed ontologies. We aimed to answer the following question: *does our context and user preference representation improve the accuracy of recommendation algorithms?*

We relied on the ConcertTweets dataset [120], which combines implicit and explicit user ratings with rich content as well as spatiotemporal contextual dimensions and social network data. It contains ratings that refer to musical shows and concerts of various artists and bands. Since the dataset was generated automatically, there were some duplicated events, for example, the same concert occurred twice, because the country appeared once as *United Kingdom* and once as *UK*. We fixed this kind of situations by eliminating the duplicates. Another problem with the dataset is the use of two rating scales: one numerical scale with ratings in the range [0.0, 5.0] and one descriptive scale with possible values equal to *yes*, *maybe* and *no*, although *no* never occurred. We decided to split the dataset into two separate sets according to the scale type and we mapped the descriptive values *yes*, *maybe* and *no* with the numerical values 2, 1 and 0. Table 6.2 presents some statistics about the data by considering the whole dataset and each of the sets generated when splitting by scale type. We prepared two pairs (one for each scale) of training and test sets for hold-out validation. In each test set, we put 20% of the newest ratings of each user. All other ratings were placed in each training set. The split was performed based on rating timestamp values.

Our pre-filtering approach can be used with existing recommendation methods. Thus, we evaluated the ontologies with five algorithms: Random Guess, Item *k*NN, User Average, SVD++ and Time SVD++. We compared the results of the first four algorithms without pre-filtering and with pre-filtering, while the fifth was executed without pre-filtering only, because it already contains time as a contextual

	All	Descriptive ratings	Numeric ratings
Number of users	61803	56519	16479
Number of musical events	116320	110207	21366
Number of pairs artist and musical events	137382	129989	23383
Number of ratings	250000	219967	30033
Maximum number of ratings per user	1423	1419	92
Minimum number of ratings per user	1	1	1
Average number of ratings per user	4.045	3.892	1.823
Maximum number of ratings per item	218	216	38
Minimum number of ratings per item	1	1	1
Average number of ratings per item	2.149	1.996	1.406
Number of users who ranked at least 5 items	13241	11548	962
Number of users who ranked at least 10 items	5369	4639	190
Number of users who ranked at least 50 items	289	244	4
Number of users who ranked at least 100 items	66	54	0
Sparsity	0.999971	0.999970	0.999922

Table 6.2 Statistics of the ConcertTweets dataset available at the time of the experiment.

factor [121]. We used it as a baseline for comparing our contextual pre-filtering technique combined with the SVD++ algorithm.

We performed an experiment for the rating prediction task and measured accuracy with MAE. Results are presented in Table 6.3 and Figures 6.12 and 6.13. It should be noticed that without pre-filtering, the User Average algorithm outperforms SVD++. This may be due to the way in which users rate musical events: it may be possible that they do not use the whole rating scale but just a part of it, e.g. a user evaluates only those events that they like (their ratings are always greater than 3.0). As can be seen in Figure 6.12 and Table 6.3, when our ontological pre-filtering approach is

	Numeric ratings		Descriptive Ratings	
Contextual pre-filtering	**YES**	**NO**	**YES**	**NO**
Random Guess	0.2315	2.0998	0.4694	0.4989
User Average	0.2312	0.3026	0.3624	0.2570
Item kNN	0.2312	0.3976	0.3624	0.4374
SVD++	0.2514	0.3511	0.3621	0.3101
Time SVD++	NA	0.2693	NA	0.2975

Table 6.3 MAE values computed for the whole test set.

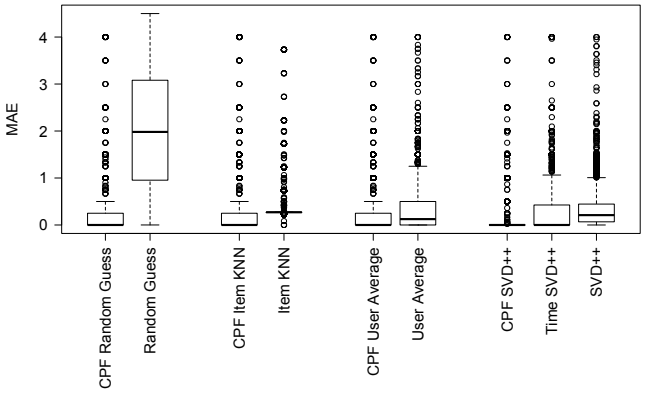


Fig. 6.12 MAE of different algorithms computed per user on subsets with numeric ratings.

applied, results on the numerical scale subset are better. Our contextual pre-filtering combined with classical SVD++ performs better than Time SVD++. There could be two reasons for this behavior: either the use of various contextual parameters in addition to time improves prediction accuracy or our approach (even if used with the time parameter only) with SVD++ is truly better than the Time SVD++ algorithm. This should be addressed in further work.

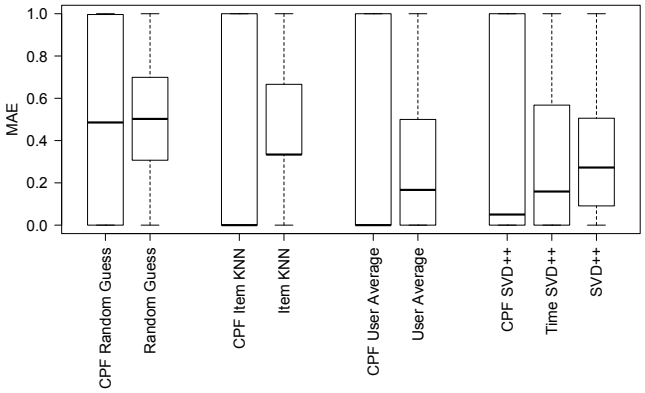


Fig. 6.13 MAE of different algorithms computed per user on subsets with descriptive ratings.

Figure 6.13 shows that the median value of MAE for our approach is lower for all algorithms, but the overall MAE for the descriptive scale subset is higher for all of the cases. This suggests that in the case of binary scale (*yes*/*maybe*) contextual pre-filtering may increase the sparsity and noisiness of the data. Thus, the recommendation algorithm may not always predict the rating. However, a user may rate differently the same event before and after participating: this could be the cause of different results for the two subsets. The rating is more reliable when a user evaluates an item after they consume it than when they declare what they would do or prefer. This could lead us to the conclusion that this approach could be successfully applied in recommender systems where numeric scale is used to rate items a posteriori. Currently, we have not identified any other limitation for using the proposed contextual pre-filtering approach.

To verify the statistical significance of the results, we applied the Wilcoxon test with *p-value* 0.01. We chose this statistical test because we cannot guarantee the normal distribution of the results obtained. The test confirmed the statistical significance of our results.

6.7 Conclusions and Future Work

In this chapter, we presented a new approach for contextual pre-filtering in recommender systems. It is based on two ontologies: Recommender System Context (RSCtx), which models the user's context, and Contextual Ontological User Profile (COUP), which represents user preferences. RSCtx extends PRISSMA and represents different context dimensions on different granularity levels. COUP was modeled according to the SIM approach for modularization. Different parts of the user profile are represented in different ontological modules. This allows us to: (i) store multiple users in one ontology, (ii) clearly distinguish user preferences from different domains keeping all the user preferences in the same ontology, and (iii) split user interests from one domain into "micro profiles" related to some contextual situation without loosing the possibility to reason on different level of context granularity.

We successfully applied RSCtx for context identification and generalization tasks, showing that it is possible to represent context by combining various dimensions and representing different granularities for each dimension. We used COUP for

representing user preferences in various contexts in the domain of musical events and for obtaining user data relevant to his current context for rating prediction task with baseline algorithms. An offline study showed that the usage of proposed ontologies with a number of recommendation algorithms can significantly improve their prediction accuracy. This confirmed part of the second research question, i.e. that it is possible to represent user preferences and items in such a way that can be combined with different recommendation approaches. The next step in our research is proving that we can adapt our user representation to various domains.

As future work, we plan to extend our experiment to ranking tasks as well as to investigate the influence of the proposed approach on diversity and novelty of recommendations. We also plan to integrate the contextual pre-filtering approach into Allied (introduced in Chapter 4). For example, since all the algorithm implemented within the framework need an initial resource to provide recommendations, the pre-filtering method could be exploited to select the initial resource from a set of user ratings based on the context.

Chapter 7

Visualizing Linked Data Based Recommendations

7.1 Introduction

DBpedia[1] is one of the main datasets in the Web of Data. It has the highest number of connections with other datasets and represents about four million resources. The data in DBpedia are extracted from Wikipedia and are available in different languages.

Linked Data resources on the Web are steadily increasing, but there is still a lack of effective ways to present them to users. Since Linked Data relies on representations and languages such as RDF [11] and SPARQL [15], dereferencing a URI in the Web of Data rarely provides an intuitive representation of the resource. In particular, there is a limited number of Linked Data browsers, and they are mainly oriented to tech-users, i.e. expert users who understand Linked Data technologies [122]. Thus, it is necessary to provide user-friendly visualizations of Linked Data and hide the complexity of RDF and SPARQL to lay-users, who do not have the skills to understand these technologies.

In this chapter, we present DBpedia Mobile Explorer, a Linked Data visualization framework for the mobile environment, which allows users to explore DBpedia by hiding the complexity of RDF and SPARQL. It can be configured as a generic DBpedia browser, thus enabling the visualization of the whole dataset, or it can focus

[1] http://dbpedia.org

on a limited number of resources and properties, generating a browser customized for a specific domain. The framework is designed to work with DBpedia, but can be adapted to other datasets and also to other services in the Web of Data, since it relies only on RDF and SPARQL.

The framework has been used in the mobile application presented in Section 5.5.1 to display recommendations based on Linked Data and originated from the needs of Telecom Italia. In effect, the company was interested in having a visualization layer that could adapt to different domains, to be used in combination with the algorithm introduced in Chapter 5 which is cross-domain.

The remainder of the chapter is organized as follows. Section 7.2 reviews the works which addressed visualization of Linked Data. Section 7.3 introduces DBpedia Mobile Explorer, particularly, it presents two reference use cases of the framework and details how it works. Finally, Section 7.4 provides the conclusions.

7.2 Linked Data Visualization

DBpedia Mobile Explorer can be considered as one of the works which try to hide the complexity of SPARQL and RDF to lay-users. In the following, we discuss a number of DBpedia interfaces and Linked Data browsers; then we review other works which provide user interfaces to deal with SPARQL and RDF.

Dadzie and Rowe [122] performed a survey on Linked Data visualization approaches. In the following, we mention the main works they considered and some additional relevant works.[2] DBpedia mobile [123] is a Linked Data browser for mobile devices. It displays nearby locations available in DBpedia on a map and allows users to browse information about them. It exploits Marbles,[3] which is a server-side application that formats Web of Data content for XHTML clients by exploiting Fresnel [124] vocabularies. DBpedia Viewer [125] is a DBpedia interface, which provides browsing and integration functions for Linked Data. It allows some actions on triples visualized (e.g. annotation) and integrates some DBpedia services and external visualization tools, such as LodLive [126] and RelFinder [127]. The former is a tool which can browse a SPARQL endpoint directly by using a JavaScript

[2] A complete review of the relevant literature is beyond the scope of this thesis. Dadzie and Rowe [122] provided a more detailed discussion of the related works.

[3] http://mes.github.io/marbles/

application layer without any application Server being needed and exploiting a graph-based visualization, while the latter enables users to explore connections among entities by showing paths in the underlying RDF graph.

There are also visualization tools not oriented to DBpedia. Payola[4] is a web framework for analyzing and visualizing Linked Data [128]. It provides an editor in which SPARQL queries and custom plugins can be combined. Furthermore, the user can choose a visualizer to see the results in various forms. Pubby[5] is a Java web application that provides a Linked Data interface to SPARQL endpoints. It presents the data available about each resource using a static HTML interface.

We also considered other works which try to hide SPARQL and RDF complexity. Shi3ld Policy Manager [129] is a user interface for querying and editing Linked Data on a SPARQL endpoint. It supports dataset administrators in defining access control policies on target elements. Linked Data Query Wizard [130] is a web-based tool for displaying, accessing, filtering, exploring, and navigating Linked Data stored in SPARQL endpoints. The main visualization functionality is converting graphs in tables. Ngomo et al. [131] presented a framework to be integrated into applications where lay-users are required to understand SPARQL or to generate SPARQL queries by converting them into natural language. Sonntag and Heim [132] provided graph visualizations and navigations of RDF resources in a mobile environment, but only in a football domain.

In summary, none of the previously mentioned works has all the characteristics that we require. In fact, we need to support lay-users and mobile devices. Additionally, it is necessary to address different domains without preventing users from focusing on a specific one. Only two of the discussed works support mobile devices, and both of them deal with information from a single domain (geographic locations or data about football).

[4] http://payola.cz/
[5] http://wifo5-03.informatik.uni-mannheim.de/pubby/

7.3 Towards a Linked Data Visualization Framework for Mobile Devices

7.3.1 Use Cases

We consider two different use cases for the framework. In both, the essential function for the end users is browsing DBpedia, but the scope may change. In fact, DBpedia is a cross domain dataset, and sometimes users may not be interested in all the information provided, on the contrary, they may want to focus on a particular domain. A user may be interested only in films, or cities, or books. Thus it may be better to exclude resources which are not relevant to avoid information overload and noise. For this reason, and considering that interest is subjective, we designed a use case to visualize any DBpedia resource and another which allows browsing only a limited number of resources. In the following, we addressed the two use cases separately.

Generic DBpedia Browser

In this use case, the user can visualize DBpedia resource, and also any property. A typical interaction of a user with the framework is the following:

1. The user input a resource available in DBpedia.
2. A brief description is presented, and a graph including all the triples having that resource as subject can be visualized. The main information about the resource (e.g. the title and director of a movie) is also reported in tabular form.
3. From the resource it is possible to browse other resources starting from its DBpedia categories. In particular, all the categories concerning the resource are retrieved, and a list is presented to the user.
4. The user can browse a category and obtain a list of all the resources included in that category. Then, he can select a resource and restart from step 2.

Domain-based DBpedia Browser

From the user's point of view, the framework behaves as reported in Section *Generic DBpedia Browser*, but it only considers resources in a particular domain. For

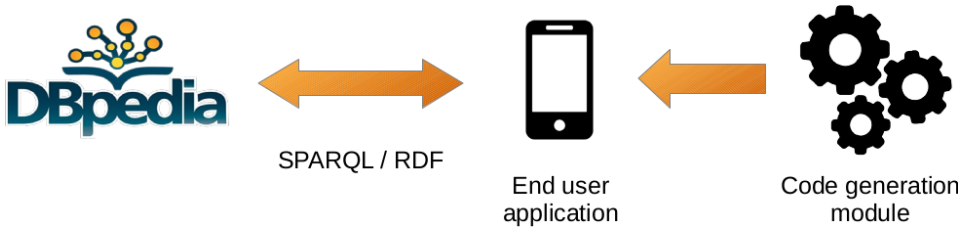


Fig. 7.1 The main components of our framework.

resources out of the domain, it is possible to visualize only a brief description and it is forbidden to browse other resources (i.e. it is impossible to obtain the categories which included a given resource and explore them). The domain of interest is not fixed and can be configured as explained in Section 7.3.2.

7.3.2 DBpedia Mobile Explorer

The framework is made up of the three modules shown in Figure 7.1. DBpedia is the dataset which provides the resources to be visualized, while the mobile application is responsible for presentation. The interaction between them is based on SPARQL and RDF. The code generation module provides part of the application. It generates the code depending on the current configuration of the framework.

The mobile application is organized according to the Model View Controller (MVC) pattern. The model consists of classes, which represent the resources considered, and a parser to load them from the resources received by DBpedia. The code generation module provides these components. Controllers and views are the core of the mobile application, and they rely only on models in such a way that it is not required to update them when models change, to have a general application to be reused with different models and parsers.

In the following, we detail the approach, provide additional information about the technology to implement the framework, and describe the user interface.

Approach

Figure 7.2 illustrates how users' operations are mapped into SPARQL queries. The queries are masked to the user by the user interface.

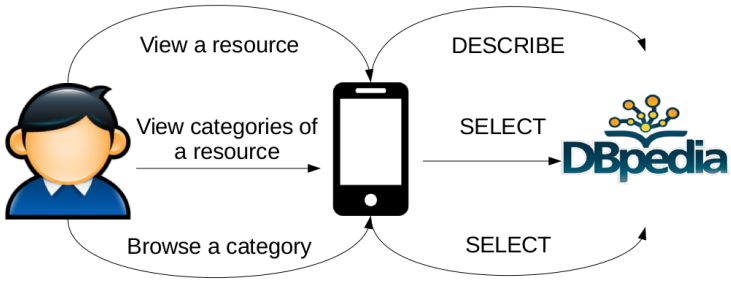


Fig. 7.2 User operations and corresponding SPARQL queries.

If DBpedia Mobile Explorer is configured as a domain-based browser, it checks the `RDF:type` property before visualizing a resource and presents only a brief description if the type does not match any of the classes specified in the domain. Similarly, the framework displays only the properties indicated. The domain of interest is custom and can be defined by specifying the classes and the properties to take into account. Any resource of a different type from the specified classes is out of the domain, and any property of a resource different from the specified properties is not considered. Tech-users or also domain experts can define the domain since it is required to know DBpedia classes and properties. This approach can be applied to any other dataset because it relies only on RDF. In fact, while defining the domain, it is possible to refer to any class or property of the underlying vocabularies.

Since the framework is designed to work with DBpedia, we exploit the `dcterms:subject` property to obtain the categories, given a resource; and also to retrieve the resources included in a given category. For referring to other category schemata, it is possible to change the property to consider. E.g. it is possible to exploit YAGO [50] classes by setting `RDF:type` instead of `dcterms:subject`. The use of `RDF:type` also enables to refer to any class, i.e. any dataset may be considered.

Technology Stack

The framework relies on an Android client application. It can interact with DBpedia using a number of SPARQL queries to access the resources and browse the categories. The RDF serialization currently supported is JSON-LD[6] and JSON is also the format to receive SPARQL results.

[6] http://json-ld.org/

```
<parserClass
    className="DBpediaParser"
    propPrefix="http://dbpedia.org/property/">
    ...
</parserClass>
<class className="Movie">
  <RDFTypes>
    <type>
     http://dbpedia.org/class/yago/Movie106613686
    </type>
    ...
  </RDFTypes>
  <listProperties>
   <prop>distributor</prop>
   <prop>director</prop>
   ...
  </listProperties>
  <dbpediaObjectProperties>
   <prop>gross</prop>
   <prop>budget</prop>
   ...
  </dbpediaObjectProperties>
  <stringProperties>
   <prop>title</prop>
  </stringProperties>
</class>
<class className="Person">
  ...
</class>
```

Listing 7.1 Example of a configuration file for the code generation module.

The code module generation is a Java application based on the CodeModel[7] library. It relies on an XML configuration file to specify the DBpedia classes and

[7] https://codemodel.java.net/

properties to consider in the generated Java classes. Listing 7.1 shows an example in which we managed movies and people which participate in them, such as actors and directors, and their related properties. The client application exploits generated Java files, which correspond to the model layer. Besides, some XML configuration files are necessary for some internal purposes. The process is summarized in Figure 7.3.

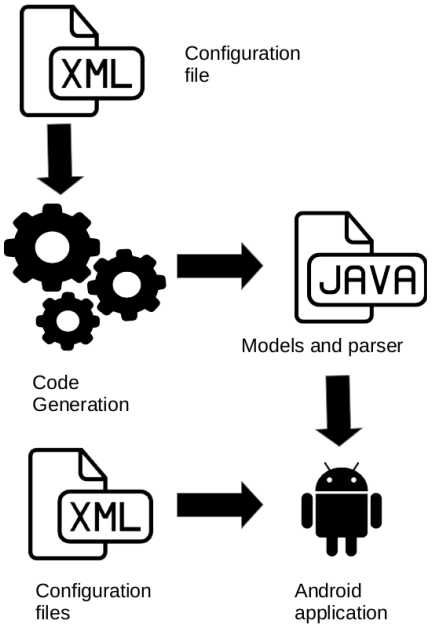
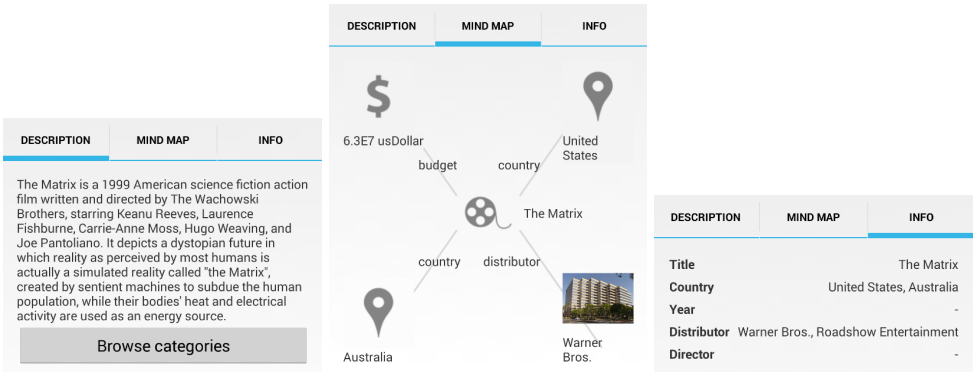


Fig. 7.3 A summary of the code generation and the configuration of our framework.

User Interface

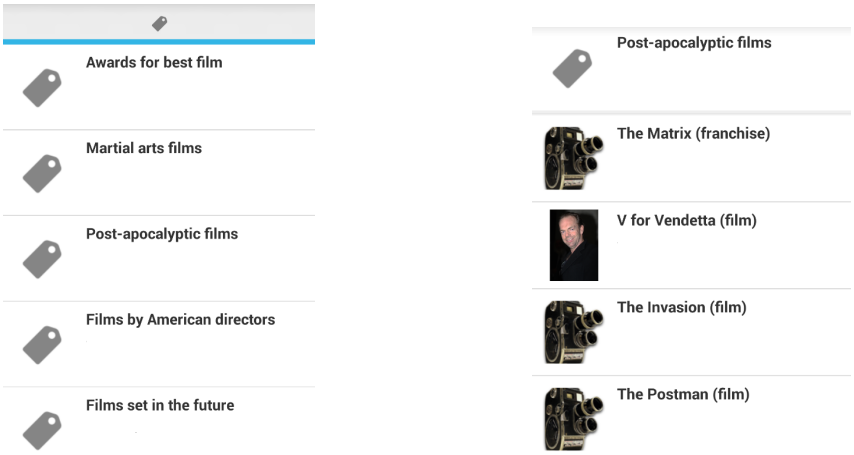
The mobile application is made up of different screens, which were designed in collaboration with psychologists specialized in user interfaces. Firstly, in the home page, the user can insert the resource to be visualized. Then a screen with three tabs is shown: (i) a description, which also allows the user to start browsing the categories of the resource, as it is illustrated in Figure 7.4a; (ii) a paginated graph view of the resource, as it is depicted in Figure 7.4b; and (iii) a tabular view with the main information about the resource (Figure 7.4c).

The graph view allows users to visualize the relationship of the currently showed resource with other resources, or just present the values of certain properties. The number of edges displayed is limited to guarantee readability since typically there are many properties for any resource. The user can scroll up or down to see other



(a) Description. (b) Graph view. (c) Main information.

Fig. 7.4 Visualizing a resource. Description, graph view and table to summarize the main information related to the resource `dbpedia:The_Matrix`.



(a) Categories of `dbpedia:The_Matrix`. (b) Resources in `Post-apocalyptic_films`.

Fig. 7.5 Browsing the categories of a resource. The categories of `dbpedia:The_Matrix` and the resources included in the category `Post-apocalyptic_films`.

properties. Thus she can explore the whole graph by focusing on a portion of the graph per time. Only the properties specified in the configuration file are displayed when the framework is configured as domain-based DBpedia browser. The same holds for the tab which presents the essential information about the resource: it is possible to specify which properties are meant to be shown; the others are excluded.

In order to browse the categories of a resource, the user is presented with the list of all the categories. When a category is tapped, all the resources which belong to it are presented and any of the resources may be visualized by tapping on it (in the

same way of the initially given resource). Figure 7.5a and Figure 7.5b respectively depict the list of categories and of resources belonging to a category.

7.4 Conclusions and Future Work

In this chapter, we addressed the problem of Linked Data visualization, especially targeting lay-users. In fact, a visualization tool for mobile devices which was not limited to a single domain was still lacking. The work originated from the needs of Telecom Italia, which was interested in displaying recommendations based on Linked Data being able to adapt to different domains.

To address this problem, we proposed DBpedia Mobile Explorer, a Linked Data visualization framework for the mobile environment. It allows users to browse DBpedia resources by hiding the complexity of RDF and SPARQL, to support lay-users. Furthermore, although it is applied to DBpedia, the approach exploited is general because it is based only on RDF and SPARQL. Thus, it could also be adopted for other datasets or services in the Web of Data.

The framework can either be configured as a general DBpedia browser or can be limited to a custom domain, by specifying the classes and properties to consider in the underlying vocabularies. The graph view allows the user to explore the whole graph by focusing on a portion of it per time, since it presents a number of properties at a time, to guarantee readability. Additionally, it is possible to browse the categories of a resource and access other resources starting from these categories.

As future work, we aim to apply the approach to other datasets and with other category systems, such as YAGO categories. Then, we plan to extend browsing. Firstly, we want to exploit direct connections among resources by mean of any properties: e.g. passing from viewing information about a movie to visualizing details on one of the actors participating in the movie, by exploiting the `dbpprop:starring` property. Secondly, we aim to obtain super categories via hierarchical properties, such as `SKOS:broader`. Finally, we are extending the framework to support other RDF serializations and other mobile operating systems.

Chapter 8

Use Cases of a Telecommunication Operator to Recommend Resources for Further Information

8.1 Introduction

The past several years have seen the Web's evolution into a Semantic Web, with a continuous increase of information published as Linked Data. This increase has generated new opportunities for annotation and categorization systems to reuse this data as semantic knowledge bases, which can be interconnected and structured to increase annotation and categorization mechanisms' precision and recall.

TellMeFirst,[1] developed at the Nexa Center for Internet & Society, is a software tool for automatically classifying and enriching documents using semantics. Here, we briefly describe some technical details about TellMeFirst and then demonstrate how a major telecommunications operator in Italy has used this software in two practical industrial cases. Looking for ways to add value to its services, Telecom Italia introduced TellMeFirst functionalities to its Society and Friend TV applications. By describing these two use cases, we seek to provide a concrete example of how research and innovation can offer advantages at the business level when applied in real commercial services. In particular, by applying TellMeFirst to these two

[1] http://tellmefirst.polito.it/

applications, we enabled the recommendation of resources from the Web of Data for providing the user with further information.

The rest of this chapter is organized as follows: Section 8.2 provides an overview about text classification and annotation; Section 8.3 introduces TellMeFirst; Section 8.4 describes Society and Friend TV. We conclude in Section 8.5.

8.2 Text Classification and Annotation

Text classification is the assignment of a text to one or more preexisting classes (also known as *features*). This process determines a text document's class membership given a set of distinct classes with a profile and various features [133]. The criterion for selecting relevant features for classification is essential and is determined a priori by the classifier (human or software). Semantic classification occurs when the classification's target elements refer to the document's meaning.

Text annotation refers to the common practice of adding information to the text itself through underlining, notes, comments, tags, or links. Text annotation can be semantic when a document's text is added with information about either the overall document's meaning or the meaning of individual elements that compose it [134]. This is done primarily using links that connect a word, expression, or phrase to an information resource on the Web or to an entity in a knowledge base [135].

8.3 TellMeFirst

The TellMeFirst project began in October 2011 at the Nexa Center for Internet & Society in Politecnico di Torino's Department of Control and Computer Engineering. The project received funding from the Working Capital-National Innovation Award. It's available under the GNU AGPLv3 license at GitHub.[2]

TellMeFirst automatically classifies and enriches documents using DBpedia[3] as the reference knowledge base for content extraction and disambiguation. Similar software tools include DBpedia Spotlight [88] and Apache Stanbol.[4] We chose

[2] http://github.com/TellMeFirst
[3] http://dbpedia.org/
[4] https://stanbol.apache.org/

DBpedia for semantic classification because the Wikipedia corpus is a perfect training set for categorization approaches based on machine learning (wherein software agents learn from data [136]) and for semantic annotation because it's directly connected to Wikipedia's vast, multilingual, pre-annotated corpus [12].

As noted, TellMeFirst exploits the relationship between Wikipedia and DBpedia to perform semantic annotation and classification processes quickly and efficiently. Although this feature distinguishes it from similar tools, it also makes it dependent on these datasets. Given the Web of data's open nature, which isn't limited to a single dataset, we must consider future evolution toward compatibility with multiple datasets.

8.3.1 Semantic Annotation

TellMeFirst's semantic annotation process associates semantic information with the words contained in a text, that is, identifying which meaning a sentence uses. This problem is well known as word-sense disambiguation (WSD). To address it, TellMeFirst provides a disambiguator that implements three subcomponents: knowledge-based, corpus-based, and first-sense heuristic disambiguators. When a term isn't disambiguated by the knowledge-based disambiguator or the corpus-based disambiguator with a certain degree of confidence, then the first-sense heuristic disambiguator assigns the most common meaning. To do this, it exploits Wikipedia resources' coefficient of prominence, i.e. the number of times each word contained in the text to be annotated is mentioned in Wikipedia through a *wikilink* (an internal link within Wikipedia). The heuristic approach is often only a few percentage points below WSD system performance [137].

We carried out tests of TellMeFirst's disambiguators on a corpus of 10 newspaper excerpts. Table 8.1 summarizes the results. The last column shows the different disambiguators' possible usage scenarios.

8.3.2 Semantic Classification

TellMeFirst implements a memory-based learning approach to semantic classification. This approach is a subcategory of the lazy learning family [138] as regards the classification phase (consultation time) and the calculation of the similarity with

Disambiguator	**Average time per word(s)**	**Average precision**	**Average recall**	**Canonical use case**
Corpus-based	0,04	0,85	0,21	Online annotation of news portals or blog
Knowledge-based	0,07	0,99	0,05	Automatic classification of documents based on DBpedia
First sense heuristic	0,04	0,78	0,24	Online annotation of news portals or blog in a more generic boundary, where the most common Wikipedia meaning is the most likely
Default	0,10	0,96	0,08	Offline annotation, automatic classification, text enhancement

Table 8.1 Tests on the TellMeFirst's Disambiguator.

the training set. A distinctive feature of the memory-based approach, also known as instance-based learning, is that the system does not create an abstract model of the classification categories (profiles) prior to text categorization. Instead, it assigns the target document to a class on the basis of a local comparison between the pre-classified documents and the target [139, 140].

The classifier must hold in memory all instances of the training set and calculate, during the classification stage, the distance vector between the training documents and the unclassified ones. An alternative approach, eager learning, conducts this operation in a learning phase (training time) in which specific category profiles are created and the function to perform the classification is defined [141].

TellMeFirst's semantic classification process is performed using the *k*-nearest neighbor (*k*NN) algorithm. This algorithm is a memory-based approach that chooses the categories to which the target document belongs based on the *k* most similar documents in a space vector [141]. The training set consists of all paragraphs in which a wikilink exists. These paragraphs are stored in an Apache Lucene[5] index: each DBpedia resource (correlated with a Wikipedia page) corresponds to a Lucene

[5] http://lucene.apache.org/

document, and for each document, there is a `CONTEXT` field for every paragraph in which the resource appears as a wikilink.

During the classification (following a lazy approach), the target document is transformed into a Boolean Lucene query over the index's `CONTEXT` field to discover the conceptual similarity with the contexts of Wikipedia entries. To calculate the similarity, TellMeFirst uses Lucene's default similarity, which combines the Boolean model with the vector space model (VSM)

Those results approved by the Boolean search on the index are then sorted according to the VSM. Lucene takes care of the stemming, lemmatization, and filtering (through specific Italian or English stop words) of the features for both the training documents and the target document transformed into a query. The query and the training documents become feature vectors (depending on the bag-of-words model), in which each feature's weight is calculated according to the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm. The query returns a list of documents (DBpedia URIs) ordered according to a similarity score that's based on the cosine similarity. This well-known metric is robust for scoring the similarity between two textual strings and is frequently used in complex queries [142]. Once TellMeFirst's similarity process obtains the ordered list of results, it applies the RCut method for thresholding [143, 144], keeping only the top seven results and discarding the others.

For classification, TellMeFirst uses a technique based on the VSM that represents both the training documents and the target document. We can view the similarity between two documents geometrically as the distance between the two vectors that represent the documents in an n-dimensional vector space, where n is the number of features in the entire training corpus. The VSM is also the basis of the Lucene libraries. Lucene is optimized to quickly calculate the distance between the documents according to the TF-IDF algorithm: given a query that represents the target document's features, it returns a list of similar documents that are indexed, even when the index contains millions of documents. The score obtained with Lucene represents the inverse of the distance between two documents: the higher the score, the closer the documents are in the vector space.

To show the results, TellMeFirst provides a visualizer containing a window with seven frames varying in size (according to the first seven previously ranked results). Each frame indicates an argument extracted from the text, and its size represents

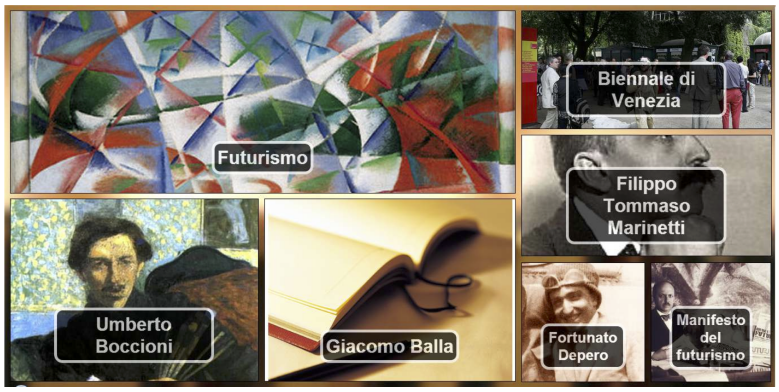


Fig. 8.1 An example of the results displayed by the TellMeFirst visualizer. The seven frames shown in this figure represent the arguments extracted from a text. The dimension of each frame indicates the relevance of each represented argument.

the text's relevance. Figure 8.1 shows an example of the results displayed in the visualizer's seven frames. [6]

8.3.3 Components of TellMeFirst

TellMeFirst is made up of the following modules:

Document parser It is used to extract the textual information from documents in diverse formats such as PDF, Word, HTML etc. It was built using libraries such as: Apache PDFBox, Apache POI and Snacktory.

Part-Of-Speech tagger and lemmatizer It is a wrapper that uses external services for text lemmatization of adjectives and common names. This module reduces errors caused by homographs of adjectives and common names.

Data extractor This module extract and processes the data obtained from DBpedia and Wikipedia. It can work offline as preprocessor.

Dictionary-based spotter This module deals with the extraction of terms from the text to annotate. It uses an internal dictionary created with the information obtained by the data extractor and extracts the terms that matches with the entries of the dictionary.

Disambiguator This module is in charge of the disambiguation of the text.

[6] Other examples are available in our demo at http://tellmefirst.polito.it

Classifier It uses a number of datasets in the Web of Data to establish the argument of the text processed. It performs SPARQL queries on DBpedia from a list of concepts from DBpedia (generated by the disambiguator) to find the entities containing the large number of links (object properties) with the concepts extracted from the input document.

Enhancer This component finds new information and new content to be added to the document to be enriched (the enhanced document) from the list of entities of DBpedia that where individuated as text argument.

Visualizer This is the display module of the system and it is responsible for collecting in a single interface the new information with which the input document has been enhanced. The results are presented in a windows with seven frames with diverse size. Each frame indicates an argument extracted from the text and its size represents its relevance according to the input document.

8.4 TellMeFirst in Practice

Telecom Italia has implemented TellMeFirst to enhance two services: Society and FriendTV.

8.4.1 Society

Society is a Telecom Italia platform that lets users in a social community share notes and comments while reading an e-book. TellMeFirst enables this service to analyze the content, notes, and comments to extract semantic concepts and hence let readers deepen the information an e-book contains.

Society is composed of a reader community that can share comments about a paragraph or even contribute to improve an e-book by sending correction reports to authors. Groups of readers can form based on social networking relationships. Users can share comments as notes in the social network, thus propagating them to other users based on a user's sharing configuration (settings). Users can share each note through a specific interface to the most-used social network, such as Facebook or Twitter. Through this interface, friends or followers can see what other users did, read their notes and add comments, or retweet the note to give more results to this

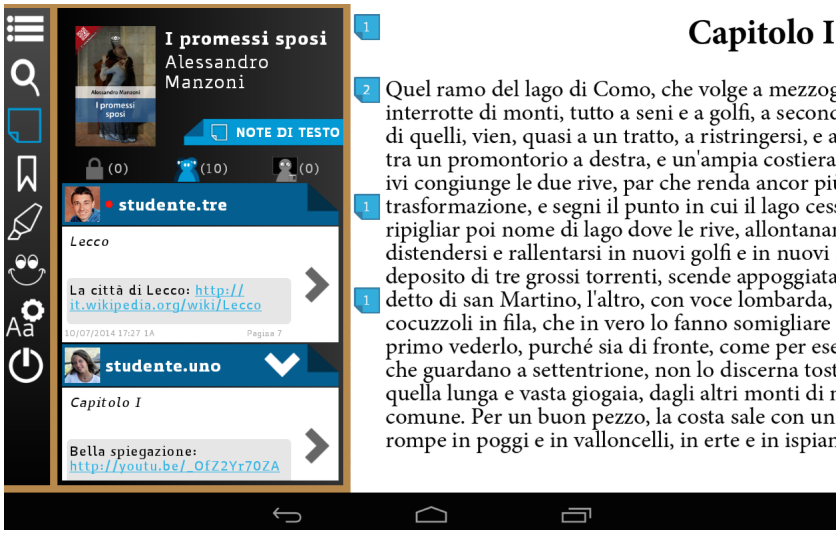


Fig. 8.2 The graphic interface of the Society application for Android devices.

piece of information. A particular interface from the social network to the system node platform can help users extract and enrich notes with information from other users on the same social network platform (Figure 8.2).

Integration with TellMeFirst is an important feature because it lets the Society application semantically annotate user-generated notes. When creating a new note or a comment on Society, TellMeFirst analyzes them (the new notes or comments) to recognize each relevant entity, such as places, names, and concepts and links them with concepts or resources in the Web of Data. The main results can then be returned, and the application can show them via the user interface, letting users save these results as a note that adds extra information to the book. Additionally, once the relevant entities are linked to resources in the Web of Data, it is possible to apply Linked Data based algorithms (as the one described ni Chapter 5) to suggest additional related information.

The Society application can be used on-the-go because it's aware of the user's context (detecting an entity as a place). In this case, the semantic source can also provide location information and be used as an extra field for searching more content, such as multimedia user-generated content that matches the same location.

Society also offers a traditional search function or feature that uses the Web as a common source for multimedia and extra information, such as images, video, audio, and text information related to words or sentences written in the book. At

the same time, the note platform can provide its information to other applications to show notes in their target interface based on location information. Moreover, the same application provides some accessibility functions that extend the book-reading experience to those with impaired abilities, for instance, by reading the text through a text-to-speech (TTS) engine for those with blindness or adjusting font sizes for those with limited vision.

This application also fits perfectly with education initiatives aimed at digitizing schools, allowing interactive education processes that avoid hard-printed books and enabling the exchange of instant messages between teachers and students.

At the moment, Telecom Italia is exploiting Society mainly as a social initiative in these two areas. However, economic benefits are also possible through an e-book distribution that supports social comment exchange features. Society is available as a mobile application for Android[7] and iOS[8] devices.

8.4.2 FriendTV

FriendTV is a social television service that lets users share TV experiences with other viewers on social media via tablets and smartphones. FriendTV presents a list of TV programs that might be of interest to a user. It uses a semantic annotator and classifier from TellMeFirst to extract and associate the concepts (based on their semantic meaning) contained in each program's description with related existing Web resources. Hence, users can easily browse for additional information about related concepts.

We can view this service as a TV guide that's integrated with Twitter and Facebook to let users discuss the most-followed TV programs on social media sites and also receive related suggestions. With FriendTV, a user can obtain information about scheduled TV programs, communicate to other users what he or she is watching, and set broadcast notifications for programs of interest. Users can also rate TV programs, letting the system provide better recommendations. Furthermore, the service lets broadcasters and media agencies release questionnaires, compute statistics based on social media, and insert banners with program information or advertisements.

[7] https://play.google.com/store/apps/details?id-it.telecomitalia.society
[8] https://itunes.apple.comlitlapp/society-school-2.0/id785451519?mt-8

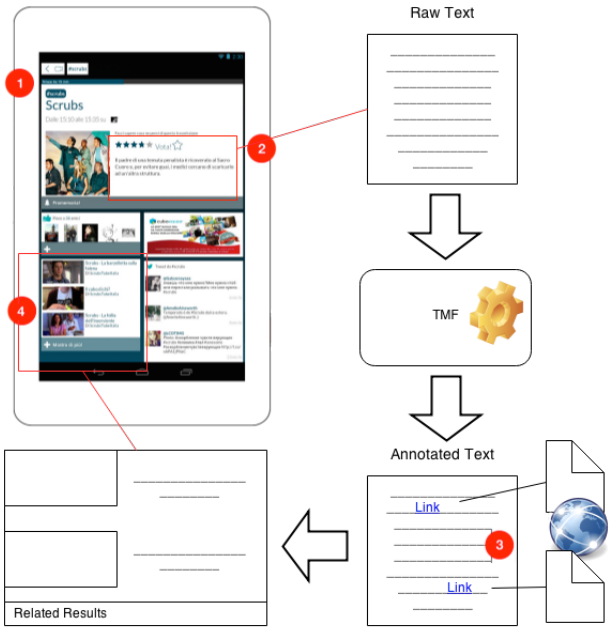


Fig. 8.3 Using TellMeFirst in FriendTV. (1) The user selects a program. (2) The raw text composing the description of the selected program is given as input to TellMeFirst, which (3) generates the annotated version of the same text, in which several entities are linked to existing and related Web resources. (4) The user receives these resources as related content.

TellMeFirst is integrated into this service to provide users with content related to a program. In fact, users can open a detailed view of a particular program and receive related content, such as videos. Thus, by starting from the TV program description, TellMeFirst can annotate the text, classify it, and exploit the links with other resources the annotation generates to retrieve semantically related content. Figure 8.3 depicts the FriendTV service's workflow. More information about FriendTV is available on the Web.[9] Additionally, FriendTV is available as a mobile application for Android[10] and iOS[11] devices. It currently counts thousands of downloads.

8.5 Conclusions and Future Work

The Web is facing a crucial challenge in promoting the construction of a new knowledge infrastructure. This is a fundamental task of Linked Data in achieving

[9] http://www.stv.telecomitalia.it/
[10] https://play.google.com/store/apps/details?id-it.telecomitalia.friendtv
[11] https://itunes.apple.com/it/app/friendtv/id784514746

the Semantic Web vision. The TellMeFirst software platform takes advantage of the information present in the Web of Data to generate semantic annotation and concept classification. To give users of this platform maximum benefit, we had to establish a joint collaboration with mobile provider Telecom Italia and mobile service users in direct contact with the Web of Data.

When users employ these services in their real lives, both TellMeFirst's creators and telecommunications operators benefit. In fact, through the increased use of semantic annotation and classification functionalities, we've detected areas of improvement. In particular, they enables the recommendation of additional information from the Web of Data. Moreover, with these new features, operators can find new ways to monetize these services and make them increasingly innovative.

As future work, we plan to improve TellMeFirst's functionalities by introducing a Linked Data based concept recommender that can suggest similar concepts related to those originally extracted in the initial semantic annotation process. This improvement will enable a whole new scenario of multidomain recommendations. Additionally, as we mentioned, efforts are under way to adapt TellMeFirst's operation with knowledge bases beyond DBpedia.

Chapter 9

Conclusions and Perspectives

9.1 Summary of Contributions

The work presented in this thesis aimed to discover useful information for the user from the huge amount of structured data, and notably Linked Data available on the Web. In particular, three main goals were defined: (i) exploiting existing relationships between resources published on the Web to provide recommendations, (ii) representing users and their context to provide better recommendations, and (iii) effectively visualizing the recommended resources and their relationships.

This thesis showed how it is possible to exploit Linked Data available on the Web to recommend useful resources to users. They have been successfully applied to recommender systems to provide cross-domain and novel recommendations about music, movies and tourist attractions, and to address well-known problems such as data sparsity and context representation. Various proposed approaches have been applied to use cases of Telecom Italia in a mobile scenario. More specifically, the main contributions of this thesis are:

- A systematic literature review summarizing the state of the art in Linked Data based RS, and suggesting further research directions, which was introduced in Chapter 3.
- The participation in the design and development of the Allied framework, presented in Chapter 4.

- ReDyAl, a new algorithm which iteratively exploits relationships among resources in the Web of Data, and its evaluation, described in Chapter 5.
- The RSCtx ontology for representing the context of a user to be provided with recommendations, its application into a new context-aware pre-filtering approach for recommender systems, and its evaluation (Chapter 6).
- DBpedia Mobile Explorer, a new visualization framework for DBpedia targeting mobile devices, which has been used to display recommendations based on Linked Data, as showed in Chapter 7.
- Application of the previously mentioned contributions to the use cases of Telecom Italia (outlined in Chapter 5 and Chapter 8), in order to suggest tourist attractions, movies, and additional information about books and TV programs, to users in a mobile context.

The systematic literature review focused on the research problems addressed and on the contributions proposed in the area of Linked Data based RS. It classified Linked Data based RS into categories according to the use of Linked Data and summarized the application domains targeted and the evaluation techniques used. The work described in this thesis is based on the outcome of this review.

Allied is a framework for the deployment and execution of Linked Data based recommendation algorithms. It also facilitates studies which evaluate them in different application domains, without being bounded to a single dataset. Thus, the framework makes it possible to benchmark the algorithms and choose the one that best fits the recommendation requirements. In the current implementation, it relies on DBpedia, but it is equally suitable to other datasets. Additionally, Allied is designed to be used as the main component for recommendations in a given architecture. In this way, developers do not need to deal with the execution platform of the algorithms but only to focus their efforts either on selecting an existing algorithm or on writing a customized one.

Although the author contributed to the development of Allied, the framework is beyond the scope of this thesis. Nonetheless, ReDyAl was integrated into the framework, and Allied was used to evaluate ReDyAl. We plan to extend the framework by incorporating the contextual pre-filtering approach, although it can only be used with collaborative filtering techniques since it relies on user preferences. Thus, it

does not apply to ReDyAl and the other Linked Data based algorithms implemented within Allied. To provide some algorithms to be used with our approach, the LibRec[1] library can be integrated.

ReDyAl is a new algorithm which relies on Linked Data by exploiting existing relationships between resources to recommend related resources. It iteratively analyzes the categories they belong to and their explicit references to other resources, then combines the results. The algorithm was tested with DBpedia, but it could as well be adapted to other datasets on the Web of Data with minor adjustments. It is not bound to any particular application domain, but can be calibrated for a given domain in order to obtain more specific results. A user study comparatively evaluated its accuracy and novelty against three state-of-the-art algorithms and showed that it provides a higher number of new recommendations while keeping a satisfying prediction accuracy.

The RSCtx OWL ontology describes the user's context for RS. In the philosophy of Linked Data, it reuses terms from third party ontologies and can be extended. It models contextual information as the sum of various dimensions on different levels of granularity, may be reused in multiple domains and applications, and complies with most common context definitions. The ontology is published on the Web according to Linked Data principles. It was used in a new contextual pre-filtering approach which can be combined with existing recommendation algorithms. An offline study evaluated the proposed approach with a rating prediction task, which showed that the use of the proposed ontology and our pre-filtering technique with some well-known recommendation algorithms significantly improved the prediction accuracy.

DBpedia Mobile Explorer is a Linked Data visualization framework for the mobile environment, which allows users to explore DBpedia by hiding the complexity of RDF and SPARQL. It can be configured as a generic DBpedia browser, thus enabling the visualization of the whole dataset, or it can focus on a limited number of resources and properties, and thus generating a browser customized for a particular domain. Our framework was designed to work with DBpedia, but can be adapted to other datasets and also to other services in the Web of Data, since it relies only on RDF and SPARQL. It was used to display recommendations based on Linked Data and it originated from the needs of Telecom Italia.

[1] http://www.librec.net/

Various contributions were applied to some use cases of Telecom Italia. The first version of ReDyAl was employed in an eTourism platform to suggest attractions and POIs to tourists, while a mobile application to recommend movies exploited a second and improved version of the algorithm. This application also utilized the DBpedia Mobile Explorer framework to visualize the recommended resources. Additionally, semantic text classification and annotation techniques were used in Society and FriendTV through TellMeFirst to suggest additional information to users about the content of books and TV programs.

9.2 Limitations

Both ReDyAl and the contextual pre-filtering method based on RSCtx are designed to be independent of the application domain, although they can be calibrated to be used in a given domain. However, only one domain was evaluated, i.e. movie for ReDyAl and music for the contextual pre-filtering method. In the case of ReDyAl, we focused on movies because Telecom Italia was interested in exploiting it in a mobile application to suggest movies, while for the contextual pre-filtering we considered concerts because of the dataset used for the evaluation. Additionally, in these domains, participants in the studies were not required to have specific skills and a large amount of data was available. Other studies should be conducted in additional domains. Also, ReDyAl could be applied to other datasets in the Web of Data, although in this work, it was used with DBpedia only. Further research should evaluate the recommendations generated using other datasets.

Our contextual pre-filtering technique was only tested considering time and location as context because we could not find evaluation datasets with additional dimensions. We should study the impact of the other contextual dimensions designed in RSCtx and, in general, we should investigate which context dimensions could be useful for recommendation scenarios, although this may strictly depend on the application domain.

Finally, ReDyAl could also be extended to consider more than one resource in input (e.g. all the resources rated positively by a user). In order to do this, ReDyAl could be executed multiple times to generate recommendations given a number of initial resources, and subsequently the results could be merged. However, this would significantly increase the response time since the algorithm relies on SPARQL

queries to discover candidate recommendations through the links among resources, which is computationally expensive. Thus, we should study how to do this taking performance into account. Another resource to consider could be the current context of the user. We should investigate how to combine ReDyAl with our pre-filtering approach, for example the latter could select an initial resource for ReDyAl from a set of user ratings based on the context.

9.3 Publications

The systematic literature review has been published in Concurrency and Computation: Practice and Experience [145]. Allied has been accepted for publication in International Journal on Semantic Web and Information Systems [146]. A preliminary version of the ReDyAl algorithm has been presented at the Seventh Conference on Internet of Things and Smart Spaces (ruSMART) [147], together with the use case in the tourism domain. The current version was introduced at the Third Workshop on New Trends in Content-Based Recommender Systems (CBRecSys) co-located with the Tenth ACM Conference on Recommender Systems (RecSys) in 2016 [148]. The RSCtx ontology and the related contextual pre-filtering method for CARS were presented at the Federated Conference on Computer Science and Information Systems (FedCSIS) in 2016 [149]. The DBpedia Mobile Explorer framework has been presented at the first IEEE International Forum on Research and Technologies for Society and Industry [150]. The application of semantic annotation and classification to the use cases of Telecom Italia has been published in IT Professional [151].

9.4 Perspectives

The Web is leaving the era of search and entering one of discovery. Search is looking for something. Discovery is finding something that we did not know existed, or we did not know how to ask for.[2] The Web is more and more driven by recommender systems, and Linked Data is a promising trend in this area. It allows recommender systems to enrich item descriptions and user profiles for different domains and can

[2] http://archive.fortune.com/magazines/fortune/fortune_archive/2006/11/27/8394347/index.htm

mitigate the effects of well-known problems, such as the new user, new item, and data sparsity. However, using such a vast amount of interlinked data poses new challenges for well-established recommendation algorithms.

This thesis outlines some enhancements achieved by exploiting the implicit knowledge represented in the Web of Data and the benefits for the users when adopting these enhancements in application scenarios. Nonetheless, further improvement is still possible. For example, discovering latent relationships among items and users could enable diversified recommendations. Diversity is a popular topic in content-based recommender systems, which usually suffer from overspecialization. Another issue which is gaining interest is mining microblogging data and text reviews. In particular, opinion mining and sentiment analysis techniques can support recommendation methods that take into account the evaluation of aspects of items expressed in text reviews. Extracting information from raw text in the form of Linked Data can ease its exploitation and the integration. Additionally, Linked Data could also be used to explain recommendations since they encode semantic information. This could be particularly useful when unknown items are proposed: the system should assist the user in the decision process, both to justify the suggestion and provide additional information that allows the user to understand the quality of the recommended item. This could increase the transparency and scrutability of the system, and the user's trust and satisfaction.

In this thesis, we showed that Linked Data based RS generates new recommendations. This is useful because users do not want to receive recommendations about items they already know about or have previously consumed. Additionally, recommending very popular items, which can be easily discovered may not be enough. For this reason, it is important to propose items that are interesting and unexpected. This is known as serendipity and should be further investigated.

Finally, a closely related research area is exploratory search. It refers to cognitive consuming search tasks such as learning or topic investigation. Exploratory search systems also recommend relevant topics or concepts. An open question not addressed in this thesis is how to leverage the data semantics richness for exploratory search.

References

[1] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook*, pages 1–35. Springer, 2011.

[2] Sarabjot Singh Anand, Patricia Kearney, and Mary Shapcott. Generating semantically enriched user profiles for web personalization. *ACM Trans. Internet Technol.*, 7(4), October 2007.

[3] Iván Cantador, Alejandro Bellogín, and Pablo Castells. A multilayer ontology-based hybrid recommendation model. *AI Commun.*, 21(2-3):203–210, April 2008.

[4] Marco Degemmis, Pasquale Lops, and Giovanni Semeraro. A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3):217–255, 2007.

[5] Stuart E. Middleton, Nigel R. Shadbolt, and David C. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, January 2004.

[6] Bamshad Mobasher, Xin Jin, and Yanzan Zhou. *Web Mining: From Web to Semantic Web: First European Web Mining Forum, EWMF 2003, Invited and Selected Revised Papers*, chapter Semantically Enhanced Collaborative Filtering on the Web, pages 57–76. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[7] Giovanni Semeraro, Marco Degemmis, Pasquale Lops, and Pierpaolo Basile. Combining learning and word sense disambiguation for intelligent user profiling. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2856–2861, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[8] Giovanni Semeraro, Pasquale Lops, Pierpaolo Basile, and Marco de Gemmis. Knowledge infusion into content-based recommender systems. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 301–304, New York, NY, USA, 2009. ACM.

[9] Cai-Nicolas Ziegler, Georg Lausen, and Lars Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *Proc. of the Thirteenth ACM Int. Conf. on Information and Knowledge Management*, CIKM '04, pages 406–415, New York, NY, USA, 2004. ACM.

[10] Tommaso Di Noia and Vito Claudio Ostuni. *Reasoning Web. Web Logic Rules: 11th Int. Summer School 2015, Berlin, Germany, July 31- August 4, 2015, Tutorial Lectures.*, chapter Recommender Systems and Linked Open Data, pages 88–113. Springer International Publishing, Cham, 2015.

[11] David Wood, Markus Lanthaler, and Richard Cyganiak. RDF 1.1 concepts and abstract syntax. W3C recommendation, W3C, February 2014. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[12] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2014.

[13] Fedelucio Narducci, Matteo Palmonari, and Giovanni Semeraro. *Cross-Language Semantic Retrieval and Linking of E-Gov Services*, pages 130–145. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[14] Gediminas Adomavicius and Alexander Tuzhilin. *Recommender Systems Handbook*, chapter Context-Aware Recommender Systems, pages 217–253. Springer US, Boston, MA, 2011.

[15] Steven Harris and Andy Seaborne. SPARQL 1.1 query language. W3C recommendation, W3C, March 2013. http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[16] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[17] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, December 1992.

[18] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 194–201, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[19] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported*

Cooperative Work, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM.

[20] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 1995)*, pages 210–217. ACM Press, 1995.

[21] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 39–46, New York, NY, USA, 2010. ACM.

[22] Guy. Shani and Asela Gunawardana. Evaluating recommendation systems. *Recommender Systems Handbook*, pages 257–297, 2011.

[23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

[24] Robin Burke. *Hybrid Web Recommender Systems*, pages 377–408. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[25] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[26] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011.

[27] Alexander Felfernig, Michael Jeran, Gerald Ninaus, Florian Reinfrank, and Stefan Reiterer. Toward the Next Generation of Recommender Systems: Applications and Research Challenges. In *Multimedia Services in Intelligent Environments*, chapter Smart Inno, pages 81 – 98. Springer, 2013.

[28] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[29] Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.

[30] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[31] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 253–260, New York, NY, USA, 2002. ACM.

[32] Xia Ning, Christian Desrosiers, and George Karypis. *A Comprehensive Survey of Neighborhood-Based Recommendation Methods*, pages 37–76. Springer US, Boston, MA, 2015.

[33] Daniele Dell'Aglio, Irene Celino, and Dario Cerizza. Anatomy of a semantic web-enabled knowledge-based recommender system. In *CEUR Workshop Proceedings*, volume 667, pages 115–130, 2010.

[34] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[35] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edition, 2011.

[36] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.

[37] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[38] Gavin Carothers and Eric Prud'hommeaux. RDF 1.1 turtle. W3C recommendation, W3C, February 2014. http://www.w3.org/TR/2014/REC-turtle-20140225/.

[39] Fabien Gandon and Guus Schreiber. RDF 1.1 XML syntax. W3C recommendation, W3C, February 2014. http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/.

[40] Dan Brickley and R.V. Guha. RDF Schema 1.1. W3C recommendation, W3C, February 2014. http://www.w3.org/TR/rdf-schema/.

[41] Barbara Kitchenham. Procedures for Performing Systematic Reviews. Technical report, Keele University, Eversleigh, Australia, 2004.

[42] Barbara Kitchenham and Stuart Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical report, University of Durham, Durham, UK, 2007.

[43] Laurent Candillier, Kris Jack, Françoise Fessant, and Frank Meyer. State-of-the-art recommender systems. *Collaborative and Social Information Retrieval and Access-Techniques for Improved User Modeling*, pages 1–22, 2009.

[44] Jesus Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46(0):109 – 132, 2013.

[45] Nicolas Marie and Fabien Gandon. Survey of linked data based exploration systems. In *Proceedings of the 3rd International Conference on Intelligent Exploration of Semantic Data - Volume 1279*, IESD'14, pages 66–77, Aachen, Germany, Germany, 2014. CEUR-WS.org.

[46] Barbara Kitchenham and Pearl Brereton. A systematic review of systematic review process research in software engineering. *Information and Software Technology*, 55(12):2049 – 2075, 2013.

[47] Gary Marchionini. Exploratory search: From finding to understanding. *Commun. ACM*, 49(4):41–46, April 2006.

[48] Daniela S. Cruzes and Tore Dybå. Recommended steps for thematic synthesis in software engineering. In *Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement*, ESEM '11, pages 275–284, Washington, DC, USA, 2011. IEEE Computer Society.

[49] Roberto Mirizzi and Tommaso Di Noia. From exploratory search to web search and back. In *Proceedings of the 3rd Workshop on Ph.D. Students in Information and Knowledge Management*, PIKM '10, pages 39–46, New York, NY, USA, 2010. ACM.

[50] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM.

[51] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[52] Michael Ley. The DBLP computer science bibliography: Evolution, research issues, perspectives. In *SPIRE*, volume 2476 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2002.

[53] Oktie Hassanzadeh and Mariano Consens. Linked movie data base. In *Workshop on Linked Data on the Web (LDOW 2009)*, 2009.

[54] Aaron Swartz. Musicbrainz: a semantic web service. *IEEE Intelligent Systems*, 17(1):76–77, Jan 2002.

[55] Alexandre Passant and Yves Raimond. Combining social music and semantic web for music-related recommender systems. In *CEUR Workshop Proceedings*, volume 405, 2008.

[56] Yinuo Zhang, Hao Wu, Vikram Sorathia, and Viktor K. Prasanna. Event recommendation in social networks with linked data enablement. In Slimane Hammoudi, Leszek A. Maciaszek, José Cordeiro, and Jan L. G. Dietz, editors, *ICEIS (2)*, pages 371–379. SciTePress, 2013.

[57] Marieke Guy, Mathieu d'Aquin, Stefan Dietze, Hendrik Drachsler, Eelco Herder, and Elisabetta Parodi. LinkedUp: Linking open data for education. *Ariadne*, 72(4), Mar 2014.

[58] Mitsopoulou Evangelia, Davide Taibi, Daniela Giordano, Stefan Dietze, Hong Qing Yu, Panagiotis Bamidis, Charalampos Bratsas, and Luke Woodham. Connecting medical educational resources to the linked data cloud: the meducator rdf schema, store and api. In *Linked Learning 2011: 1st International Workshop on eLearning Approaches for the Linked Data Age, 8th Extended Semantic Web Conference (ESWC2011)*, 2011.

[59] Claus Stadler, Jens Lehmann, Konrad Höffner, and Sören Auer. Linkedgeodata: A core for a web of spatial open data. *Semantic Web Journal*, 3(4):333–354, 2012.

[60] Guan-Shuo Mai, Yu-Hwang Wang, Yue-Joe Hsia, Sheng-Shan Lu, and Chau-Chin Lin. Linked Open Data of Ecology (LODE): A new approach for ecological data sharing. *Taiwan Journal of Forest Science*, 26(4):417–424, Dec 2011.

[61] Joeran Beel, Stefan Langer, Marcel Genzmehr, Bela Gipp, Corinna Breitinger, and Andreas Nürnberger. Research paper recommender system evaluation: a quantitative literature survey. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, pages 15–22. ACM, 2013.

[62] Stephan Baumann, Rafael Schirru, and Bernhard Streit. Towards a storytelling approach for novel artist recommendations. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6817 LNCS, pages 1–15, 2011.

[63] Ulrich Küster and Birgitta König-Ries. Measures for benchmarking semantic web service matchmaking correctness. In *The Semantic Web: Research and Applications*, pages 45–59. Springer, 2010.

[64] Marjorie McShane, Sergei Nirenburg, and Stephen Beale. NLP with reasoning and for reasoning. In Chu-ren Huang, Nicoletta Calzolari, Aldo Gangemi, Alessandro Lenci, Alessandro Oltramari, and Laurent Prevot, editors, *Ontology and the Lexicon: A Natural Language Processing Perspective*, chapter Ontology,, pages 98–121. Cambridge University Press, Cambridge, 2010.

[65] Jacopo Urbani, Jason Maassen, Niels Drost, Frank Seinstra, and Henri Bal. Scalable rdf data compression with mapreduce. *Concurrency and Computation: Practice and Experience*, 25(1):24–39, 2013.

[66] Alexandre Passant. dbrec - Music Recommendations Using DBpedia. In *The Semantic Web - ISWC 2010*, pages 209–224. Springer Berlin Heidelberg, 2010.

[67] Xavier Amatriain, Josep M. Pujol, and Nuria Oliver. I like it... i like it not: Evaluating user ratings noise in recommender systems. In *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization: Formerly UM and AH*, UMAP '09, pages 247–258, Berlin, Heidelberg, 2009. Springer-Verlag.

[68] Marco Rossetti, Fabio Stella, and Markus Zanker. Contrasting offline and online results when evaluating recommendation algorithms. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 31–34, New York, NY, USA, 2016. ACM.

[69] Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. Adoption of the linked data best practices in different topical domains. In *The Semantic Web - ISWC 2014*, volume 8796 of *Lecture Notes in Computer Science*, pages 245–260. Springer International Publishing, 2014.

[70] Alistair Miles and Sean Bechhofer. SKOS simple knowledge organization system reference. W3c recommendation, W3C, 2009.

[71] Danica Damljanovic, Milan Stankovic, and Philippe Laublet. Linked data-based concept recommendation: Comparison of different methods in open innovation scenario. 7295:24–38, 2012.

[72] Milan Stankovic, Werner Breitfuss, and Philippe Laublet. Discovering Relevant Topics Using DBPedia: Providing Non-obvious Recommendations. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pages 219–222. IEEE, August 2011.

[73] Nuno Seco, Tony Veale, and Jer Hayes. An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In *European Conference on Artificial Intelligence*, pages 1089–1090, 2004.

[74] Mohamed Ali Hadj Taieb, Mohamed Ben Aouicha, Mohamed Tmar, and Abdelmajid Ben Hamadou. New information content metric and nominalization relation for a new wordnet-based method to measure the semantic relatedness. In *Cybernetic Intelligent Systems (CIS), 2011 IEEE 10th International Conference on*, pages 51–58, Sept 2011.

[75] Yutaka Kabutoya, Robert Sumi, Tomoharu Iwata, and Toshio Tadasu Uchiyama. A topic model for recommending movies via linked open data. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pages 625–630. IEEE, December 2012.

[76] Erik Mannens, Sam Coppens, Toon De Pessemier, Hendrik Dacquin, Davy Van Deursen, Robbie De Sutter, and Rik Van de Walle. Automatic news recommendations via aggregated profiling. *Multimedia Tools and Applications*, 63(2):407–425, 2013.

[77] Vito Claudio Ostuni, Tommaso Di Noia, Eugenio Di Sciascio, and Roberto Mirizzi. Top-N recommendations from implicit feedback leveraging linked open data. In *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13*, RecSys '13, pages 85–92. ACM Press, 2013.

[78] Ladislav Peska and Peter Vojtas. Enhancing Recommender System with Linked Open Data. In *10th International Conference on Flexible Query Answering Systems (FQAS 2013)*, pages 483–494, Granada, Spain, 2013. Springer Berlin / Heidelberg.

[79] Iván Cantador, Ioannis Konstas, and Joemon M. Jose. Categorising social tags to improve folksonomy-based recommendations. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(1):1–15, March 2011.

[80] Hussam Hamdan. Experiments with DBpedia , WordNet and SentiWordNet as re- sources for sentiment analysis in micro-blogging. In *Seventh International Workshop on Semantic Evaluation (SemEval 2013) - Second Joint Conference on Lexical and Computational Semantics*, volume 2, pages 455–459, Atlanta, Georgia, 2013.

[81] Koki Kitaya, Hung-Hsuan Huang, and Kyoji Kawagoe. Music Curator Recommendations Using Linked Data. In *Second International Conference on the Innovative Computing Technology (INTECH 2012)*, pages 337–339. IEEE, September 2012.

[82] Houda Khrouf and Raphaël Troncy. Hybrid event recommendation using linked data and user diversity. In *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13*, RecSys '13, pages 185–192. ACM Press, 2013.

[83] Gong Cheng, Saisai Gong, and Yuzhong Qu. An Empirical Study of Vocabulary Relatedness and Its Application to Recommender Systems. In *10th International Conference on The Semantic Web - Volume Part I*, pages 98–113. Springer, 2011.

[84] Victor de Graaff, Anne van de Venis, Maurice van Keulen, and Rolf A. de By. Generic knowledge-based analysis of social media for recommendations. In *CBRecSys 2015: New trends on content-based recommender systems*. CEUR-WS.org, Sept 2015.

[85] Cataldo Musto, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Simone Rutigliano. Automatic selection of linked open data features in graph-based recommender systems. In *CBRecSys 2015: New trends on content-based recommender systems*, pages 10–13. CEUR-WS.org, Sept 2015.

[86] Joseph L. Fleiss et al. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[87] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[88] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.

[89] William E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research*, pages 354–359, 1990.

[90] Krzysztof Goczyla, Wojciech Waloszek, and Aleksander Waloszek. Contextualization of a DL knowledge base. In *Proc. of the 2007 Int. Workshop on Description Logics (DL2007)*, 2007.

[91] Umberto Panniello, Alexander Tuzhilin, Michele Gorgoglione, Cosimo Palmisano, and Anto Pedone. Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 265–268, New York, NY, USA, 2009. ACM.

[92] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161 – 180, 2010. Context Modelling, Reasoning and Management.

[93] Cristiana Bolchini, Carlo A. Curino, Elisa Quintarelli, Fabio A. Schreiber, and Letizia Tanca. A data-oriented survey of context models. *SIGMOD Rec.*, 36(4):19–26, December 2007.

[94] Reto Krummenacher and Thomas Strang. Ontology-based context modeling. In *In Workshop on Context-Aware Proactive Systems*, 2007.

[95] Juan Ye, Lorcan Coyle, Simon Dobson, and Paddy Nixon. Ontology-based models in pervasive computing systems. *Knowl. Eng. Rev.*, 22(4):315–347, December 2007.

[96] Luca Costabello. *Context-Aware Access Control and Presentation for Linked Data*, chapter A Declarative Model for Mobile Context, pages 21–32. 2013.

[97] Harry Chen, Tim Finin, and Anupam Joshi. *Ontologies for Agents: Theory and Experiences*, chapter The SOUPA Ontology for Pervasive Computing, pages 233–258. Birkhäuser Basel, Basel, 2005.

[98] Thomas Strang, Claudia Linnhoff-Popien, and Korbinian Frank. *Distributed Applications and Interoperable Systems: 4th IFIP WG6.1 Int. Conf., DAIS 2003, Paris, France, November 17-21, 2003. Proc.*, chapter CoOL: A Context Ontology Language to Enable Contextual Interoperability, pages 236–247. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[99] Xiao Hang Wang, Da Qing Zhang, Tao Gu, and Hung Keng Pung. Ontology based context modeling and reasoning using owl. In *Proc. of the Second IEEE Annual Conf. on Pervasive Computing and Communications Workshops*, PERCOMW '04, pages 18–, Washington, DC, USA, 2004. IEEE Computer Society.

[100] Davy Preuveneers, Jan Bergh, Dennis Wagelaar, Andy Georges, Peter Rigole, Tim Clerckx, Yolande Berbers, Karin Coninx, Viviane Jonckers, and Koen Bosschere. *Ambient Intelligence: Second European Symposium, EUSAI 2004, Eindhoven, The Netherlands, November 8-11, 2004. Proc.*, chapter Towards an Extensible Context Ontology for Ambient Intelligence, pages 148–159. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[101] Panu Korpipää and Jani Mäntyjärvi. *Modeling and Using Context: 4th Int. and Interdisciplinary Conf. CONTEXT 2003 Stanford, CA, USA, June 23–25, 2003 Proc.*, chapter An Ontology for Mobile Device Sensor-Based Context Awareness, pages 451–458. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[102] Ramón Hervás and José Bravo. Towards the ubiquitous visualization: Adaptive user-interfaces based on the semantic web. *Interacting with Computers*, 23(1):40 – 56, 2011.

[103] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Proc. of the 1st Int. Symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, London, UK, UK, 1999. Springer-Verlag.

[104] Marius Kaminskas and Francesco Ricci. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, 6(2–3):89 – 119, 2012.

[105] Rui Cai, Chao Zhang, Chong Wang, Lei Zhang, and Wei-Ying Ma. Musicsense: Contextual music recommendation using emotional allocation modeling. In *Proc. of the 15th ACM Int. Conf. on Multimedia*, MM '07, pages 553–556, New York, NY, USA, 2007. ACM.

[106] Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci. Context relevance assessment and exploitation in mobile recommender systems. *Personal Ubiquitous Comput.*, 16(5):507–526, June 2012.

[107] Zhenglian Su, Jun Yan, Haifeng Ling, and Haisong Chen. Research on personalized recommendation algorithm based on ontological user interest model. *J. of Computational Information Systems*, 8(1):169–181, January 2012.

[108] Christian Rack, Stefan Arbanowski, and Stephan Steglich. Context-aware, Ontology-based Recommendations. In *SAINT-W '06: Proc. of the Int. Symposium on Applications on Internet Workshops*, pages 98–104, Washington, DC, USA, 2006. IEEE Computer Society.

[109] Iván Cantador, Alejandro Bellogín, and Pablo Castells. Ontology-based personalised and context-aware recommendations of news items. In *Proc. of the 2008 IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '08, pages 562–565, Washington, DC, USA, 2008. IEEE Computer Society.

[110] José Rodríguez, Maricela Bravo, and Rafael Guzmán. Multidimensional ontology model to support context-aware systems, 2013.

[111] Ahmad Hawalah and Maria Fasli. Utilizing contextual ontological user profiles for personalized recommendations. *Expert Systems with Applications*, 41(10):4777 – 4797, 2014.

[112] Anind K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7, January 2001.

[113] M. Fernández-López, A. Gómez-Pérez, and N. Juristo. Methontology: from ontological art towards ontological engineering. In *Proc. Symposium on Ontological Engineering of AAAI*, 1997.

[114] Dominik Heckmann, Tim Schwartz, Boris Brandherm, Michael Schmitz, and Margeritta Wilamowitz-Moellendorff. *User Modeling 2005: 10th Int. Conf., UM 2005, Edinburgh, Scotland, UK, July 24-29, 2005. Proc.*, chapter Gumo – The General User Model Ontology, pages 428–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[115] James A. Russell. A circumplex model of affect. *J. of Personality and Social Psychology*, 39(6):1161–1178, December 1980.

[116] Albert Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292.

[117] Krzysztof Goczyla, Aleksander Waloszek, and Wojciech Waloszek. Towards context-semantic knowledge bases. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *Federated Conf. on Computer Science and Information Systems - FedCSIS 2012, Wroclaw, Poland, 9-12 September 2012, Proc.*, pages 475–482, 2012.

[118] Krzysztof Goczyła, Aleksander Waloszek, Wojciech Waloszek, and Teresa Zawadzka. *Intelligent Tools for Building a Scientific Information Platform*, chapter Modularized Knowledge Bases Using Contexts, Conglomerates and a Query Language, pages 179–201. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[119] Aleksandra Karpus and Krzysztof Goczyla. A multi-domain hybrid recommender systems based on a dynamic contextual ontological user profile. In *Doctoral Consortium (IC3K 2014)*, pages 83–87, 2014.

[120] Panagiotis Adamopoulos and Alexander Tuzhilin. Estimating the Value of Multi-Dimensional Data Sets in Context-based Recommender Systems. In *8th ACM Conf. on Recommender Systems (RecSys 2014)*, 2014.

[121] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '09, pages 447–456, New York, NY, USA, 2009. ACM.

[122] Aba-Sah Dadzie and Matthew Rowe. Approaches to visualising linked data: A survey. *Semant. web*, 2(2):89–124, April 2011.

[123] Christian Becker and Christian Bizer. Dbpedia mobile: A location-enabled linked data browser. In Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee, editors, *LDOW*, volume 369 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[124] Emmanuel Pietriga, Christian Bizer, David Karger, and Ryan Lee. Fresnel: A browser-independent presentation vocabulary for rdf. In Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and LoraM. Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 158–171. Springer Berlin Heidelberg, 2006.

[125] Denis Lukovnikov, Dimitris Kontokostas, Claus Stadler, Sebastian Hellmann, and Jens Lehmann. Dbpedia viewer - an integrative interface for dbpedia leveraging the dbpedia service eco system. In *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014*, 2014.

[126] Diego Valerio Camarda, Silvia Mazzini, and Alessandro Antonuccio. Lodlive, exploring the web of data. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, pages 197–200, New York, NY, USA, 2012. ACM.

[127] Philipp Heim, Steffen Lohmann, and Timo Stegemann. Interactive relationship discovery via the semantic web. In Lora Aroyo, Grigoris Antoniou, Eero Hyvönen, Annette ten Teije, Heiner Stuckenschmidt, Liliana Cabral, and Tania Tudorache, editors, *The Semantic Web: Research and Applications*, volume 6088 of *Lecture Notes in Computer Science*, pages 303–317. Springer Berlin Heidelberg, 2010.

[128] Jakub Klímek, Jiří Helmich, and Martin Nečaský. Payola: Collaborative linked data analysis and visualization framework. In Philipp Cimiano, Miriam Fernández, Vanessa Lopez, Stefan Schlobach, and Johanna Völker, editors, *The Semantic Web: ESWC 2013 Satellite Events*, volume 7955 of *Lecture Notes in Computer Science*, pages 147–151. Springer Berlin Heidelberg, 2013.

[129] Luca Costabello, Serena Villata, Iacopo Vagliano, and Fabien Gandon. Assisted policy management for sparql endpoints access control. In Eva Blomqvist and Tudor Groza, editors, *International Semantic Web Conference (Posters & Demos)*, volume 1035 of *CEUR Workshop Proceedings*, pages 33–36. CEUR-WS.org, 2013.

[130] Patrick Höfler, Michael Granitzer, Eduardo E. Veas, and Christin Seifert. Linked data query wizard: A novel interface for accessing SPARQL endpoints. In *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014*, 2014.

[131] Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, Christina Unger, Jens Lehmann, and Daniel Gerber. Sorry, i don't speak sparql: Translating sparql queries into natural language. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 977–988, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[132] Daniel Sonntag and Philipp Heim. A constraint-based graph visualisation architecture for mobile semantic web interfaces. In Bianca Falcidieno, Michela Spagnuolo, Yannis Avrithis, Ioannis Kompatsiaris, and Paul Buitelaar, editors, *Semantic Multimedia*, volume 4816 of *Lecture Notes in Computer Science*, pages 158–171. Springer Berlin Heidelberg, 2007.

[133] V.K. Singh, R. Piryani, A. Uddin, P. Waila, and Marisha. Sentiment analysis of textual reviews; evaluating machine learning, unsupervised and sentiwordnet approaches. In *Knowledge and Smart Technology (KST), 2013 5th International Conference on*, pages 122–127, Jan 2013.

[134] David Sánchez, David Isern, and Miquel Millan. Content annotation for the semantic web: an automatic web-based approach. *Knowledge and Information Systems*, 27(3):393–418, 2011.

[135] Alexander Hogenboom, Frederik Hogenboom, Flavius Frasincar, Kim Schouten, and Otto van der Meer. Semantics-based information extraction for detecting economic events. *Multimedia Tools and Applications*, 64(1):27–52, 2013.

[136] Ron Kohavi and Foster Provost. Glossary of terms. *Machine Learning*, 30:271–274, 1998.

[137] Diana McCarthy. Word sense disambiguation: An overview. *Language and Linguistics Compass*, 3(2):537–558, 2009.

[138] Aastha Gupta, Rachna Rajput, Richa Gupta, and Monika Arora. Hybrid model to improve time complexity of words search in pos tagging, Sept 2014.

[139] Weiwei Cheng and Eyke Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009.

[140] Bin Fu, Zhihai Wang, Guandong Xu, and Longbing Cao. Multi-label learning based on iterative label propagation over graph. *Pattern Recognition Letters*, 42(0):85 – 90, 2014.

[141] Claude Sammut and Geoffrey I. Webb, editors. *Encyclopedia of Machine Learning*. Springer, 2010.

[142] David C. Anastasiu and George Karypis. L2ap: Fast cosine similarity search with prefix l-2 norm bounds. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 784–795, March 2014.

[143] Yiming Yang. A study of thresholding strategies for text categorization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 137–145, New York, NY, USA, 2001. ACM.

[144] Xiaofeng He, Rong Zhang, and Aoying Zhou. Threshold selection for classification with skewed class distribution. In Yunjun Gao, Kyuseok Shim, Zhiming Ding, Peiquan Jin, Zujie Ren, Yingyuan Xiao, An Liu, and Shaojie Qiao, editors, *Web-Age Information Management*, volume 7901 of *Lecture Notes in Computer Science*, pages 383–393. Springer Berlin Heidelberg, 2013.

[145] Cristhian Figueroa, Iacopo Vagliano, Oscar Rodríguez Rocha, and Maurizio Morisio. A systematic literature review of linked data-based recommender systems. *Concurrency and Computation: Practice and Experience*, 2015.

[146] Cristhian Figueroa, Iacopo Vagliano, Oscar Rodriguez Rocha, Marco Torchiano, Catherine Faron-Zucker, Juan Carlos Corales, and Maurizio Morisio. Allied: A framework for executing linked data-based recommendation algorithms. *International Journal on Semantic Web and Information Systems*, 13(3), In press.

[147] Oscar Rodriguez Rocha, Cristhian Figueroa, Iacopo Vagliano, and Boris Moltchanov. Linked data-driven smart spaces. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems - 14th International Conference, NEW2AN 2014 and 7th Conference, ruSMART 2014, St. Petersburg, Russia, August 27-29, 2014. Proceedings*, pages 3–15, 2014.

[148] Iacopo Vagliano, Cristhian Figueroa, Oscar Rodriguez Rocha, Marco Torchiano, Catherine Faron-Zucker, and Maurizio Morisio. Redyal: A dynamic recommendation algorithm based on linked data. In *Proceedings of the 3rd Workshop on New Trends in Content-Based Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys 2016), Boston, MA, USA, September 16, 2016.*, pages 31–38, 2016.

[149] Aleksandra Karpus, Iacopo Vagliano, Krzysztof Goczyla, and Maurizio Morisio. An ontology-based contextual pre-filtering technique for recommender systems. In *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdańsk, Poland, September 11-14, 2016.*, pages 411–420, 2016.

[150] Iacopo Vagliano, Marco Marengo, and Maurizio Morisio. DBpedia Mobile Explorer. In *1st International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, pages 181–185, Sept 2015.

[151] Oscar Rodriguez Rocha, Iacopo Vagliano, Cristhian Figueroa, Federico Cairo, Giuseppe Futia, Carlo A. Licciardi, Marco Marengo, and Federico Morando. Semantic annotation and classification in practice. *IT Professional*, 17(2):33–39, Mar 2015.

[152] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. A generic semantic-based framework for cross-domain recommendation. In *Proceedings of the 2Nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, page 25–32, New York, NY, USA, 2011. ACM.

[153] Jörg Waitelonis and Harald Sack. Towards exploratory video search using linked data. *Multimedia Tools and Applications*, 59(2):645–672, July 2012.

[154] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, page 1–8, New York, NY, USA, 2012. ACM.

[155] Rouzbeh Meymandpour and Joseph G. Davis. Recommendations using linked data. In *Proceedings of the 5th Ph.D. Workshop on Information and Knowledge*, PIKM '12, pages 75–82, New York, NY, USA, 2012. ACM.

[156] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, and Davide Romito. Exploiting the web of data in model-based recommender systems. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, page 253–256, New York, NY, USA, 2012. ACM.

Appendix A

Selection and Synthesis in the Systematic Literature Review

A.1 Initial set of papers

Table A.1 Initial set of papers and keywords listed in each of them. The search string used in the systematic review was built based on these keywords.

Paper	Keywords
Passant [66]	Semantic Web Applications, Linked Data, Recommendation Systems, Semantic Distance, DBpedia
Fernandez-Tobias et al. [152]	Recommender Systems, Cross-Domain Recommendation, Knowledge Extraction, Semantic Networks, Linked Data, DBpedia
Waitelonis and Sack [153]	Linked Open Data, Video search, Exploratory Search
Damljanovic et al. [71]	Concept Recommendation, Structure-Based Similarity, Semantic Similarity, Information Retrieval, Statistical Semantics, Linked Data, Ontologies, Recommender Systems, Concept Discovery, Open Innovation
Di Noia et al. [154]	Content-Based Recommender Systems, Vector Space Model, Linked Data, DBpedia, LinkedMDB, Freebase, Semantic Web, MovieLens, Precision, Recall
Meymandpour and Davis [155]	Linked Data, Semantic Web, Similarity Metrics, Web of Data, Recommendation, Partitioned Information Content
Di Noia et al. [156]	Model-based Recommender Systems, SVM, Linked Data, DBpedia, LinkedMDB, Semantic Web, MovieLens, Precision, Recall

A.2 Selected Papers

Table A.2 Selected papers (*P*) and corresponding studies (*S*). Rows in italics identify papers belonging to a study already reported by other paper (e.g. papers 10, 19, 54 belong to the same study S10).

P	S	Authors	Year	Title	Publication details
1	S1	Fernández-Tobías, I., Cantador, I., Kaminskas, M., Ricci, F.	2011	A generic semantic-based framework for cross-domain recommendation	2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems - HetRec '11, pp 25 - 32
2	S2	Kabutoya, Y., Sumi, R., Iwata, T., Uchiyama, T., Uchiyama, T.	2012	A Topic Model for Recommending Movies via Linked Open Data	International Conferences on Web Intelligence and Intelligent Agent Technology, pp 625 – 630
3	S3	Dell'Aglio, D., Celino, I., Cerizza, D.	2010	Anatomy of a Semantic Web-enabled Knowledge-based Recommender System	4th international workshop Semantic Matchmaking and Resource Retrieval in the Semantic Web, at the 9th International Semantic Web Conference, pp 115 – 130
4	S4	Mannens, E., Coppens, S., Wica, I., Dacquin, H., Van De Walle, R.	2013	Automatic News Recommendations via aggregated Profiling	Journal Multimedia Tools and Applications, 63 (2), pp 407 – 425
5	S5	Dzikowski, J., Kaczmarek, M.	2012	Challenges in Using Linked Data within a Social Web Recommendation Application to Semantically Annotate and Discover Venues	International Cross Domain Conference and Workshop, pp 360 – 374
6	S6	Wardhana, A.T.A.; Nugroho, H.T.	2013	Combining FOAF and Music Ontology for Music Concerts Recommendation on Facebook Application	Conference on New Media Studies, pp 1 – 5
7	S7	Passant, A., Raimond, Y.	2008	Combining Social Music and Semantic Web for music-related recommender systems	First Workshop on Social Data on the Web, pp 19 -30
8	S8	Lindley, A., Graf, R.	2011	Computing Recommendations for Long Term Data Accessibility basing on Open Knowledge and Linked Data	5th ACM Conference on Recommender Systems, pp 51 – 58

9	S9	Passant, Alexandre	2010	dbrec — Music Recommendations Using DBpedia	The Semantic Web – ISWC 2010, pp 209 – 224
10	S10	Stankovic, M., Breitfuss, W., Laublet, P.	2011	Discovering Relevant Topics Using DBPedia: Providing Non-obvious Recommendations	2011 International Conferences on Web Intelligence and Intelligent Agent Technology, 1, pp 219 - 222
11	S11	Marie, N., Gandon, F., Ribière, M., Rodio, F.	2013	Discovery Hub : on-the-fly linked data exploratory search	9th International Conference on Semantic Systems, pp 17 – 24
12	S12	Peska, L., Vojtas, P.	2013	Enhancing Recommender System with Linked Open Data	10th International Conference on Flexible Query Answering Systems, pp 483 – 494
13	S13	Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D.	2012	Exploiting the web of data in model-based recommender systems	6th ACM conference on Recommender systems
14	S14	Golbeck, J.	2006	Filmtrust: movie recommendations from semantic web-based social networks	3rd IEEE Consumer Communications and Networking Conference, pp 1314 – 1315
15	S15	Celma, Ò., Serra, X.	2008	FOAFing the music: Bridging the semantic gap in music recommendation	Web Semantics: Science, Services and Agents on the World Wide Web, 6 (4), 250 – 256
16	S16	Varga, B., Groza, A.	2011	Integrating DBpedia and SentiWordNet for a tourism recommender system	7th International Conference on Intelligent Computer Communication and Processing, pp 133 – 136
17	S17	Kaminskas, M., Fernández-Tobías, I., Ricci, F., Cantador, I.	2012	Knowledge-based music retrieval for places of interest	Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies - MIRUM '12, pp 19 – 24
18	S18	Dietze, S.	2012	Linked Data as facilitator for TEL recommender systems in research & practice	2nd Workshop on Recommender Systems for Technology Enhanced Learning, pp 7 – 10
19	*S10*	*Damljanovic, D., Stankovic, M., Laublet, P.*	*2012*	*Linked Data-Based Concept Recommendation : Comparison of Different Methods*	*9th Extended Semantic Web Conference, pp 24 – 38*

20	S19	Kitaya, K., Huang, H. H., Kawagoe, K.	2012	Music curator recommendations using linked data	Second International Conference on the Innovative Computing Technology, pp 337 – 339
21	S20	Jung, K., Hwang, M., Kong, H., Kim, P.	2005	RDF Triple Processing Methodology for the Recommendation System Using Personal Information	International Conference on Next Generation Web Services Practices, pp 241 – 246
22	S21	Calì, A., Capuzzi, S., Dimartino, M. M., Frosini, R.	2013	Recommendation of Text Tags in Social Applications Using Linked Data	ICWE 2013 Workshops
23	*S21*	*Calì, A., Capuzzi, S., Dimartino, M. M., Frosini, R.*	*2013*	*Recommendation of Text Tags Using Linked Data*	*3rd International Workshop on Semantic Search Over the Web, pp 1 – 3*
24	S22	Meymandpour, R., Davis, J. G.	2012	Recommendations using linked data	5th Ph.D. workshop on Information and knowledge - PIKM '12, pp 75 – 82
25	S23	Harispe, S., Ranwez, S., Janaqi, S., Montmain, J.	2013	Semantic Measures Based on RDF Projections: Application to Content-Based Recommendation Systems	On the Move to Meaningful Internet Systems: OTM 2013 Conferences SE – 44, pp 606 – 615
26	S24	Hopfgartner, F., Jose, J. M.	2010	Semantic user profiling techniques for personalised multimedia recommendation	Multimedia Systems, 16 (4-5), pp 255 – 274
27	*S5*	*Łazaruk, S., Dzikowski, J., Kaczmarek, M., Abramowicz, W.*	*2012*	*Semantic Web Recommendation Application*	Federated Conference on Computer Science and Information Systems (FedCSIS), pp 1055 – 1062
28	S25	Ostuni, V. C., Di Noia, T., Di Sciascio, E., Mirizzi, R.	2013	Top-N recommendations from implicit feedback leveraging linked open data	Proceedings of the 7th ACM conference on Recommender systems, pp 85 – 92
29	S26	Ahn, J., Amatriain, X.	2010	Towards Fully Distributed and Privacy-Preserving Recommendations via Expert Collaborative Filtering and RESTful Linked Data	International Conference on Web Intelligence and Intelligent Agent Technology, pp 66 – 73
30	S27	Heitmann, B., Hayes, C.	2010	Using Linked Data to Build Open , Collaborative Recommender Systems	AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, pp 76 – 81

31	S28	Zarrinkalam, F., Kahani, M.	2012	A multi-criteria hybrid citation recommendation system based on linked data	2nd International eConference on Computer and Knowledge Engineering (ICCKE), 2012, pp 283 – 288
32	S29	Lommatzsch, A., Kille, B., Kim, J. W., Albayrak, S.	2013	An Adaptive Hybrid Movie Recommender based on Semantic Data	10th Conference on Open Research Areas in Information Retrieval, pp 217 – 218
33	S30	Torres, D., Skaf-Molli, H., Molli, P.; Díaz, A.	2013	BlueFinder: Recommending Wikipedia Links Using DBpedia Properties	5th Annual ACM Web Science Conference, pp 413 – 422
34	S31	Ostuni, V. C., Di Noia, T., Mirizzi, R., Romito, D., Di Sciascio, E.	2012	Cinemappy : a Context-aware Mobile App for Movie Recommendations boosted by DBpedia	International Workshop on Semantic Technologies meet Recommender Systems & Big Data SeRSy 2012, pp 37 - 48
35	S33	Zhang, Y. Wu, H. Sorathia, V., Prasanna, V. K.	2008	Event recommendation in social networks with linked data enablement	15th International Conference on Enterprise Information Systems, pp 371 – 379
36	S34	Mirizzi, R., Di Noia, T.	2010	From exploratory search to web search and back	3rd workshop on Ph.D. students in information and knowledge management - PIKM '10, pp 39 – 46
37	S35	Khrouf, H., Troncy, R.	2013	Hybrid event recommendation using linked data and user diversity	Proceedings of the 7th ACM conference on Recommender systems, pp 185 – 192
38	S36	Bahls, D., Scherp, G., Tochtermann, K., Hasselbring, W.	2012	Towards a Recommender System for Statistical Research Data	2nd International Workshop on Semantic Digital Archives
39	S37	Cheng, Gong; Gong, Saisai; Qu, Yuzhong	2011	An Empirical Study of Vocabulary Relatedness and Its Application to Recommender Systems	10th International Conference on The Semantic Web - Volume Part I, pp 98 – 113
40	S38	Wang, Y., Stash, N., Aroyo, L., Gorgels, P., Rutledge, L., Schreiber, G.	2008	Recommendations based on semantically enriched museum collections	Web Semantics: Science, Services and Agents on the World Wide Web, 6 (4), 283 – 290
41	*S11*	*Marie, N., Gandon, F., Legrand, D., Ribière, M.*	*2013*	*Discovery Hub: a discovery engine on the top of DBpedia*	*3rd International Conference on Web Intelligence, Mining and Semantics*

42	S31	Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., Zanker, M.	2012	Linked open data to support content-based recommender systems	8th International Conference on Semantic Systems
43	*S31*	*Ostuni, Vito Claudio; Gentile, Giosia; Noia, Tommaso Di; Mirizzi, Roberto; Romito, Davide; Sciascio, Eugenio Di*	*2013*	*Mobile Movie Recommendations with Linked Data*	*International Cross-Domain Conference, pp 400 – 415*
44	*S31*	*Mirizzi, R., Di Noia, T., Ragone, A., Ostuni, V. C., Di Sciascio, E.*	*2012*	*Movie recommendation with DBpedia*	*3rd Italian Information Retrieval Workshop, pp 101 – 112*
45	S39	Waitelonis, J., Sack, H.	2011	Towards exploratory video search using linked data	Multimedia Tools and Applications, 59 (2), pp 645 – 672
46	S40	Li, S., Zhang, Y., Sun, H.	2010	Mashup FOAF for Video Recommendation LightWeight Prototype	7th Web Information Systems and Applications Conference, pp 190 – 193
47	S41	Hu, Y., Wang, Z., Wu, W., Guo, J., Zhang, M.	2010	Recommendation for Movies and Stars Using YAGO and IMDB	12th International Asia-Pacific Web Conference, pp 123 – 129
48	S42	Ruotsalo, T., Haav, K., Stoyanov, A., Roche, S., Fani, E., Deliai, R., Mäkelä, E., Kauppinen, T., Hyvönen, E.	2013	SMARTMUSEUM: A mobile recommender system for the Web of Data	Web Semantics: Science, Services and Agents on the World Wide Web, 20, pp 50 – 67
49	S43	Stankovic, M., Jovanovic, J., Laublet, P.	2011	Linked Data Metrics for Flexible Expert Search on the Open Web	8th Extended Semantic Web Conference, pp 108 – 123
50	S44	Ozdikis, O., Orhan, F., Danismaz, F.	2011	Ontology-based recommendation for points of interest retrieved from multiple data sources	International Workshop on Semantic Web Information Management, pp 1 – 6
51	S45	Debattista, J., Scerri, S., Rivera, I., Handschuh, S.	2012	Ontology-based rules for recommender systems	International Workshop on Semantic Technologies meet Recommender Systems & Big Data, pp 49 – 60
52	S46	Codina, V.; Ceccaroni, L.	2010	Taking Advantage of Semantics in Recommendation Systems	2010 Conference on Artificial Intelligence Research and Development, pp 163 – 172

53	*S9*	*Passant, A., Decker, S.*	*2010*	*Hey! Ho! Let's Go! Explanatory Music Recommendations with dbrec*	*7th Extended Semantic Web Conference, pp 411 – 415*
54	*S10*	*Stankovic, M., Breitfuss, W., Laublet, P.*	*2011*	*Linked-data based suggestion of relevant topics*	*7th International Conference on Semantic Systems, pp 49 – 55*
55	*S9*	*Passant, A.*	*2010*	*Measuring semantic distance on linking data and using it for resources recommendations*	*AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, pp 93 – 98*
56	*S14*	*Golbeck, J.*	*2006*	*Generating Predictive Movie Recommendations from Trust in Social Network*	*4th International Conference, iTrust 2006, pp 93 – 104*
57	*S39*	*Sack, H.*	*2009*	*Augmenting Video Search with Linked Open Data*	*International Conference on Semantic Systems, pp 550 – 558*
58	S47	Baumann, S., Schirru, R., Streit, B.	2011	Towards a Storytelling Approach for Novel Artist Recommendations	8th International Workshop, AMR 2010, Linz, Austria, August 17-18, 2010, Revised Selected Papers, pp 1 – 15
59	S48	Corallo, A., Lorenzo, G., Solazzo, G.	2006	A Semantic Recommender Engine Enabling an eTourism Scenario	10th International Conference, pp 1092 – 1101
60	S49	Nuzzolese, A. G., Presutti, V., Gangemi, A., Musetti, A., Ciancarini, P.	2013	Aemoo: Exploring Knowledge on the Web	Proceedings of the 5th Annual ACM Web Science Conference, pp 272 – 275
61	*S49*	*Musetti, A., Nuzzolese, A., Draicchio, F., Presutti, V., Blomqvist, E., Gangemi, A., Ciancarini, P.*	*2012*	*Aemoo: Exploratory Search based on Knowledge Patterns over the Semantic Web*	*Semantic Web Challenge*
62	*S47*	*Baumann, S., Schirru, R.*	*2012*	*Using Linked Open Data for Novel Artist Recommendations*	*13th Internal Society for Music Information Retrival Conference*
63	S50	Cantador, I., Castells, P.	2006	Multilayered Semantic Social Network Modeling by Ontology-Based User Profiles Clustering: Application to Collaborative Filtering	Proceedings of 15th International Conference, pp 334 – 349

64	*S34*	*Mirizzi, R., Ragone, A., Di Noia, T., Di Sciascio, E.*	*2010*	*Ranking the Linked Data: The Case of DBpedia*	*10th International Conference, pp 337 – 354*
65	S51	Heitmann, B.,Hayes, C.	2010	Enabling Case-Based Reasoning on the Web of Data	The WebCBR Workshop on Reasoning from Experiences on the Web at International Conference on Case-Based Reasoning
66	S52	Alvaro, G., Ruiz, C., Córdoba, C., Carbone, F., Castagnone, M., Gómez-Pérez, J. M., Contreras, J.,	2011	miKrow : Semantic Intra-enterprise Micro-Knowledge Management System	8th Extended Semantic Web Conference, pp 154 – 168
67	*S50*	*Cantador, I., Castells, P., Bellogín, A.*	*2011*	*An Enhanced Semantic Layer for Hybrid Recommender Systems: Application to News Recommendation*	*Int. J. Semant. Web Inf. Syst., 7 (1), pp 44 – 78*
68	S32	Cantador, I., Konstas, I., Jose, J. M.	2011	Categorising social tags to improve folksonomy-based recommendations	Web Semantics: Science, Services and Agents on the World Wide Web, 9 (1), pp 1 – 15
69	*S29*	*Lommatzsch, A., Kille, B., Albayrak, S.*	*2013*	*A Framework for Learning and Analyzing Hybrid Recommenders based on Heterogeneous Semantic Data Categories and Subject Descriptors*	*10th Conference on Open Research Areas in Information Retrieval, pp 137 – 140*

A.3 Excluded Papers

Table A.3 Papers (*P*) excluded from the systematic literature review during the data extraction.

P	Authors	Year	Title	Publication details	Reason
18	Wu, C., Wu, J., Ye, G., He, L., Huang, L., Xie, M.	2013	Linked Course Data-based User Personal Knowledge Recommendation Engine Architecture of Linked Course Data	Journal of Computational Information Systems, 9 (5), pp 1735 – 1742	The study is not a RS.

36	Pereira Nunes, B. P., Dietze, S., Casanova, M., A., Kawase, R., Fetahu, B., Nejdl, W.	2013	Combining a Co-occurrence-Based and a Semantic Measure for Entity Linking	10th International Conference, pp 548 – 562	The study is not a RS.
41	Ruotsalo, T., Hyvönen, E.	2007	A Method for Determining Ontology-Based Semantic Relevance	18th International Conference, pp 680 – 688	The study is not a RS. It is a semantic similarity method.
43	Parundekar, R., Oguchi, K.	2012	Driver recommendations of POIs using a semantic content-based approach	International Workshop on Semantic Technologies Meet Recommender Systems and Big Data	The study is a RS but use classical kNN algorithm for recommendation. RDF data used only in an intermediate step to integrate data.
50	Song, T., Zhang, D., Shi, X., He, J.,Kang, Q.	2014	Combining Fusion and Ranking on Linked Data for Multimedia Recommendation	Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Volume 3, pp 531-538	The full paper of the study not available
51	Pham, X.H., Jung, J.J., Takeda, H.	2013	Exploiting linked open data for attribute selection on recommendation systems	Find out how to access preview-only content	The full paper of the study not available
53	Kim, T., Kim, P., Lee, S., Jung, H., Sung, W. K.	2013	OntoURIResolver: Resolution and recommendation of URIs Published in LOD	Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Volume 3	The study is not a RS.
54	Kurz, T., Bürger, T., Sint, R., Mika, P., Vallet, D., Carrero, F. M.	2010	R3-A related resource recommender	Lecture Notes in Electrical Engineering Volume 272, 2014, pp 531-538	The study is not a RS. It is more an interlinking framework

56	Morshed, A., Dutta, R., Aryal, J.	2013	Recommending environmental knowledge as linked open data cloud using semantic machine learning	29th International Conference on Data Engineering Workshops	The study is a knowledge base based on data integration and knowledge recommendation. Linked Data is not used to provide recommendations, but only for data integration.
58	Wu, J.Y., Wu, C.L.	2014	The Study of User Model of Personalized Recommendation System Based on Linked Course Data	Applied Mechanics and Materials, 519-520, pp 1609-1612	The full paper of the study not available (excluded before data extraction)
60	Kim, T., Kim, P., Lee, S., Jung, H., Sung, W. K.	2011	OntoURIResolver: URI Resolution and Recommendation Service Using LOD	Future Generation Information Technology Conference, pp 245 – 250	The study is not a RS.
61	Bianchini, Devis	2012	A Classification of Web API Selection Solutions over the Linked Web	2nd International Workshop on Semantic Search over the Web	The study is not a RS.
69	Bellekens, P., Houben, G.-J., Aroyo, L., Schaap, K., Kaptein, A.	2009	User Model Elicitation and Enrichment for Context-sensitive Personalization in a Multiplatform Tv Environment	Proceedings of the seventh european conference on European interactive television conference, pp 119 – 128	The study is not a RS.
71	Zarrinkalam, F., Kahani, M.	2011	Improving bibliographic search through dataset enrichment using Linked Data	1st International eConference on Computer and Knowledge Engineering, pp 254 – 259	The study is not a RS. It uses Linked Data to enrich data.
72	Sheng, H., Chen, H., Yu, T., Feng, Y.	2010	Linked data based semantic similarity and data mining	2010 IEEE International Conference on Information Reuse & Integration, pp 104 – 108	The study is not a RS. It is a semantic similarity method.
73	Qing, H., Dietze, S., Giordano, D., Taibi, D., Kaldoudi, E., Dovrolis, N.	2012	Linked education : interlinking educational resources and the web of data	27th Annual ACM Symposium on Applied Computing, pp 366 – 371	The study is not a RS. It uses Linked Data to enrich data.

74	Albertoni, R., De Martino, M.	2006	Semantic Similarity of Ontology Instances Tailored on the Application Context	OTM Confederated International Conferences, CoopIS, DOA, GADA, and ODBASE 2006. Proceedings, Part I, pp 1020 – 1038	The study is not a RS. It is a semantic similarity method.
77	Tous, R., Delgado, J.	2006	A Vector Space Model for Semantic Similarity Calculation and OWL Ontology Alignment	17th International Conference, pp 307 – 316	The study is not a RS. It is a semantic similarity method.
83	Leal, J. P., Rodrigues, V., Queirós, R.	2012	Computing Semantic Relatedness using DBPedia	1st Symposium on Languages, Applications and Technologies, pp 133 – 147	The study is not a RS. It is a semantic similarity method.
85	Golbeck, J., Hendler, J.	2006	FilmTrust: movie recommendations using trust in web-based social networks	Consumer Communications and Networking Conference, pp 282 – 286	The study is a RS but not use Linked Data to provide recommendations
87	Xie, M.	2011	Semantic-Based Linked Data Mining and Services	Journal of Information and Computational Science, 12 (December), pp 3981 -3988	The study is not a RS.
90	Shabir, N., Clarke, C.	2009	Using Linked Data as a basis for a Learning Resource Recommendation System	1st International Workshop on Semantic Web Applications for Learning and Teaching Support in Higher Education	The study is not a RS.

A.4 Thematic Synthesis

The main topics covered by our systematic review are presented in the model of higher-order themes in Figure A.1. It resembles the coding that we used in the synthesis phases described in Section 3.2.3. As it can be seen, topics are grouped by each research question (RQ) proposed in Section 3.2.1.

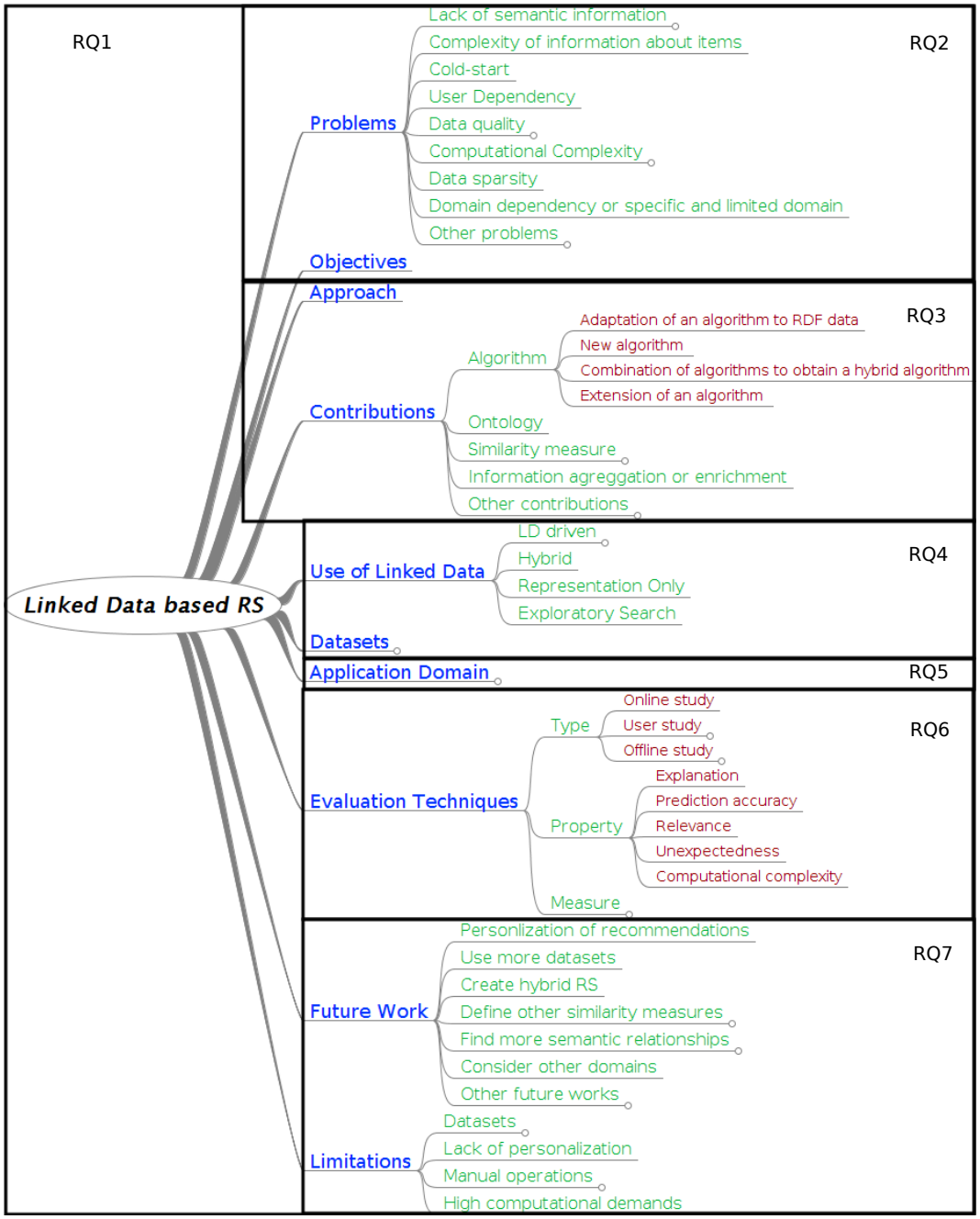


Fig. A.1 The model of higher-order themes of our systematic review.